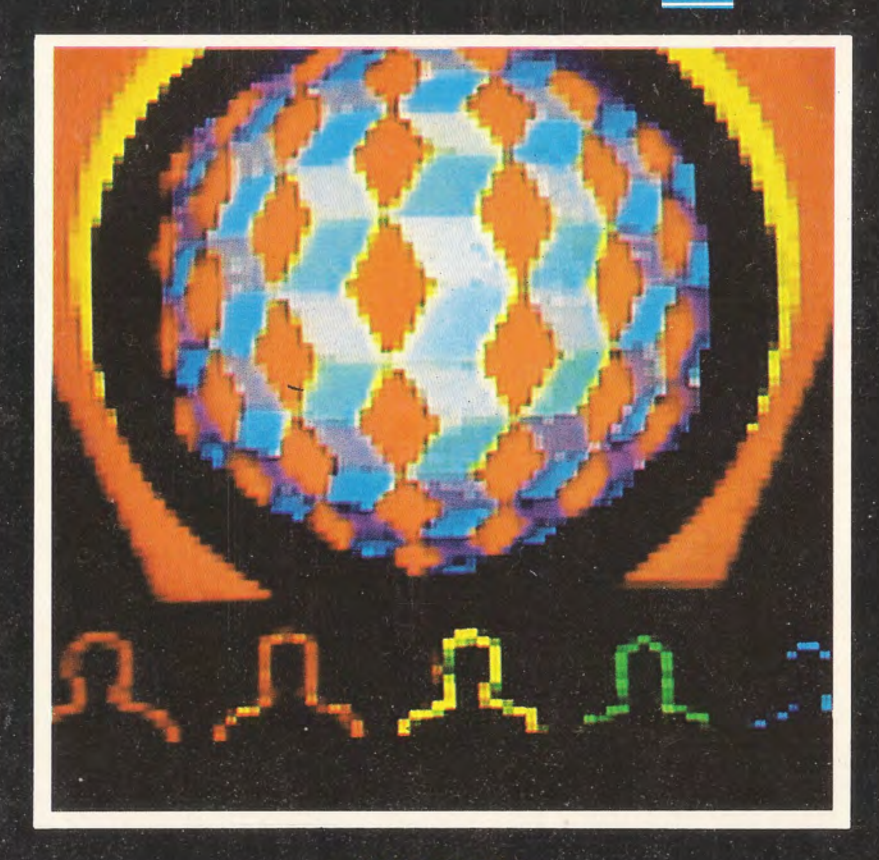
BIBLIOTECA BASICA INFORMATICA

MUSICA Y SONIDO EN SU ORDENADOR 30



INGELEK

BIBLIOTECA BASICA INFORMATICA

MUSICA Y SONIDO EN SU ORDENADOR 30

INGELEK

Director editor:
Antonio M. Ferrer Abelló.

Director de producción:
Vicente Robles.

Coordinador y supervisión técnica:
Enrique Monsalve.

Redactor técnico:
Alfredo B. García Pérez.

Colaboradores:
Casimiro Zaragoza.

Diseño:
Bravo/Lofish.

Dibujos:
José Ochoa.

ISBN del tomo: 84-85831-75-6
ISBN de la obra: 84-85831-31-4
Fotocomposición Pérez Díaz, S. A.
Imprime: Héroes, S. A.
Depósito legal: M-16947-1986
Precio en Canarias, Ceuta y Melilla: 380 pts.

INDICE

PROLOGO

No nos detendremos mucho en hablar del pasado y porvenir de la unión entre música e informática, ya que cualquiera puede ver (y sobre todo oír) sus efectos cada día. Hubo un tiempo en que la informática musical se reducía a ambientes aislados como laboratorios de investigación, pero ésta ha ido poco a poco penetrando en nuestras casas junto con los microordenadores.

Y no hay que olvidar tampoco el irresistible ascenso de la música "informatizada" de unos años a esta parte... El microprocesador se está convirtiendo en parte permanente del material musical de nuestros días (sintetizadores, baterías electrónicas, cajas de ritmos...), evolución simbolizada por el omnipresente interface MIDI.

En la actualidad casi toda la música que escuchamos (¡voluntariamente o no!) ha sufrido de una forma u otra una elaboración digital. "¡Por desgracia!", estará pensando más de un lector. Bueno, hay que reconocer que la situación actual es un poco paradójica: el uso de medios informáticos ofrece una potencia y libertad en la creación musical que no habíamos tenido nunca. Por otra parte, esos mismos medios, utilizados tan a menudo con el fin de obtener una creación fácil e inmediata (y el mayor provecho...), conducen a unos resultados marcados por una desoladora estandarización, aunque lo cierto es que aquí, como en otras ocasiones, el culpable es el hombre, no la máquina...

Volviendo a consideraciones que tocan más de cerca a los usuarios de micros domésticos, queremos comentar otra paradoja. La música es uno de los campos de la microinformática donde

se pueden obtener los resultados más satisfactorios gracias a las posibilidades ofrecidas por la última generación de ordenadores. Hagamos tan sólo una comparación con los gráficos: la resolución y los colores disponibles en una típica pantalla gráfica no pueden compararse con las ilustraciones de un libro o revista bien impresa (sin intentar siquiera hacer una comparación con los medios de que dispone un pintor....); los generadores sonoros son, en cambio, cada vez más potentes y sus prestaciones a veces no distan de las de un material profesional.

Sin embargo, son pocos los usuarios de microordenadores que logran obtener elaboraciones musicales satisfactorias. ¿A qué se debe esto? Podríamos culpar tal vez al interface del usuario, que no es muy favorable: el manejo del joystick o del ratón resulta fácil e intuitivo cuando se trata de dibujar, pero no es lo mismo cuando lo que se desea es componer música. Hay un segundo obstáculo: los comandos musicales a disposición del programador son generalmente complejos y no estandarizados. No hay que olvidar tampoco que el proceso de creación y percepción sonora obedece a ciertas reglas que no son de fácil comprensión.

Con este libro queremos ayudar en el aprovechamiento de la potencialidad de un ordenador. Debido a la extensión del tema no se puede pretender mucho de unas pocas páginas, pero esperamos que el lector encuentre en ellas un punto de partida suficiente.

El libro está estructurado en dos partes, la primera más teórica y la segunda más práctica, con programas y ejemplos diversos.

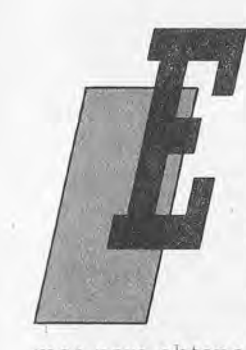
Los capítulos 1 y 2 contienen nociones esenciales acerca del análisis de un fenómeno musical y sobre los medios electrónicos que permiten su reproducción. Tales nociones son indispensables para aprovechar plenamente las posibilidades del ordenador; por ello, y aunque se requiera algo de fuerza de voluntad, es aconsejable leerlos con detenimiento (sobre todo el primero).

El capítulo 3 ofrece una breve panorámica acerca de las capacidades musicales de los ordenadores de hoy en día.

Los restantes capítulos presentan breves programas de aplicación para varios ordenadores.

CAPITULO I

ANATOMIA DE UN SONIDO



En este capítulo no trataremos directamente de música computerizada, sino que expondremos brevemente los principios del análisis de un sonido musical. Estas nociones teóricas tal vez no resultan demasiado sencillas, pero merece la pena hacer un esfuerzo y leerlo todo hasta el final, ya que comprender la influencia de los distintos parámetros sobre el sonido producido nos permitirá actuar correctamente sobre los mismos para obtener del ordenador el resultado que queremos.

Antes de nada veamos qué ocurre cuando captamos un sonido: como es sabido, el oído humano transforma las vibraciones del aire en señales que nuestro cerebro analiza y reconoce como un sonido. Si éste resulta ser musical puede ser descrito mediante tres características simples, que serán precisamente los parámetros utilizados para crear un sonido con el ordenador.

Altura de una nota musical

La primera característica es la ALTURA (o tono). Todo el mundo sabe que al pulsar una tecla de la parte derecha del teclado de un piano se produce un sonido diferente —más agudo— que si se tocara una a la izquierda. Cada nota musical corresponde a una altura bien definida. En física la magnitud que mide el tono es la FRECUENCIA del sonido; cuanto más rápida sea la vibración transmitida a nuestro oído, más agudo será el sonido correspondiente.

Un ejemplo práctico de esto se da cuando ponemos a 45 r.p.m.

un disco de 33 r.p.m.: el disco gira con mayor rapidez, las vibraciones transmitidas a la aguja y posteriormente amplificadas poseen un ritmo más rápido y escuchamos así una melodía, no sólo más veloz, sino también desplazada en cuanto al tono se refiere hacia los agudos.

La frecuencia se mide en Hertzios (abreviado Hz). Un Hertzio corresponde a una vibración completa (ciclo) por segundo. El oído humano es capaz de captar sonidos con una frecuencia comprendida generalmente entre 20 Hz y 18.000 Hz. El "LA" del diapasón posee una frecuencia de 440 Hz, que corresponde a 440 vibraciones por segundo. Por tanto, cada vibración elemental dura alrededor de 2,3 milisegundos (1000/400 = 2,3...). A esta duración se le llama *período* del fenómeno.

Es conveniente saber que cuando se pasa de una nota a su homónima situada una octava más alta (por ejemplo, al pasar de un DO al DO sucesivo del teclado) la frecuencia se duplica: la cuerda del DO más agudo efectúa dos oscilaciones en el tiempo que la otra hace sólo una.

Sin entrar en más detalles podemos señalar que las notas musicales tradicionales (las de la escala utilizada en la música occidental) poseen unas frecuencias cuyas relaciones se acercan mucho a fracciones matemáticas simples. Este cálculo justifica la colocación aparentemente arbitraria de las diferentes notas dentro de una octava.

Por otra parte, no hay nada que impida, siempre que el medio electrónico lo permita, generar sonidos con frecuencias distintas de las notas habituales y crear así diferentes escalas o frecuencias variando con continuidad en un intervalo.

El volumen

Pasemos ahora a la segunda característica de un sonido musical, que es muy fácil de entender: es el VOLUMEN (o intensidad) del sonido. Cuanto más fuerte sea el sonido, mayor será su volumen. Ahora bien, lo que nos interesa no es tanto la intensidad del sonido en valor absoluto, que, por otra parte, puede ser regulada con un mundo llamado precisamente "volumen", sino las relaciones entre los distintos volúmenes sonoros, pues serán importantes como parámetros de nuestras manipulaciones.

Tercera característica: el timbre

Queda una última característica, no tan sencilla de explicar. Una misma nota y con igual volumen se percibirá de forma diferente según el tipo de instrumento que la produzca. Se dice entonces que tales instrumentos poseen distinto TIMBRE.

Vamos a tratar de explicar algo mejor en qué difieren estos timbres y cómo nuestro oído distingue sin equivocarse una flauta de una guitarra o de una tetera que silba. Esto resulta necesario si queremos tratar de reproducir los timbres de instrumentos reales o crear otros nuevos.

La primera diferencia estriba en lo que se llama la FORMA DE ONDA (WAVEFORM en inglés). Al hablar de la frecuencia de un sonido dijimos que es el número de vibraciones por segundo, pero el término "vibración" fue utilizado de una manera un tanto imprecisa. Un sonido musical es un caso particular de un fenómeno periódico. Para conocer dicho fenómeno no basta saber con qué velocidad se producen las vibraciones, sino que también es necesario conocer el "aspecto" que presenta la onda durante un período (ya que se repetirá en toda su duración). Este "aspecto" puede ser ilustrado por medio de gráficos en los cuales está representada la amplitud de la onda en función del tiempo.

En tales gráficos (Figs. 1 a la 6) el tiempo se lee sobre el eje horizontal (el intervalo visible corresponde a un período) y la ordenada del punto representa la amplitud o alejamiento de la posición de reposo. Los ejemplos que aparecen corresponden a formas de onda frecuentes en los fenómenos musicales. Al final del capítulo volveremos a este tema para profundizar en algún aspecto importante del mismo.

Ahora bien, la forma de onda tampoco es suficiente para caracterizar el timbre de un instrumento. Un papel importante lo juega la ENVOLVENTE del sonido (ENVELOPE en inglés); describe

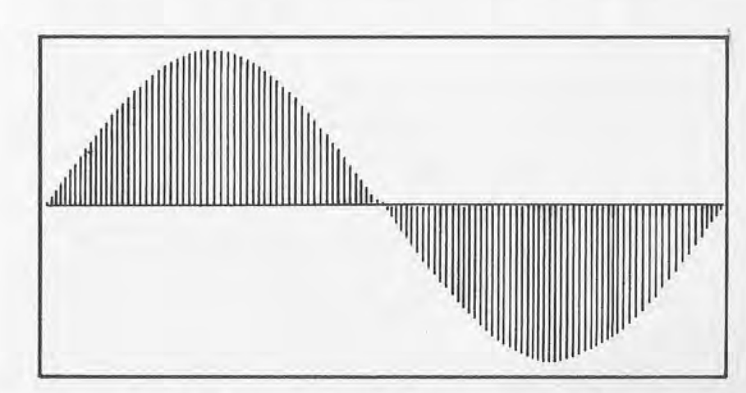


Figura 1.—Onda sinusoidal. Un instrumento real no produce nunca una onda de este tipo; el sonido correspondiente se reconoce en seguida como producido electrónicamente.

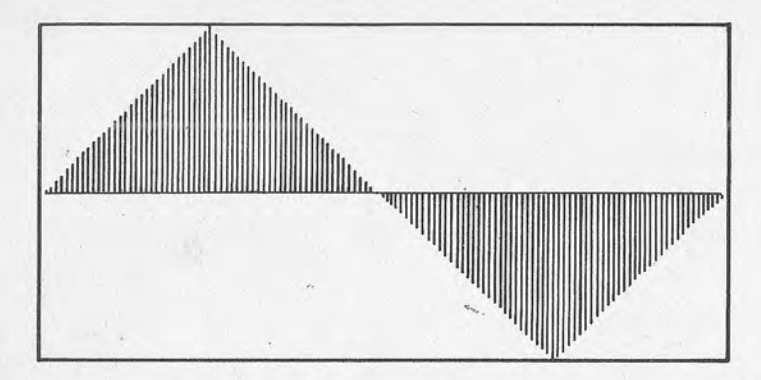


Figura 2.—Onda triangular. Se trata de una onda bastante pobre en armónicos; utilizada en los sintetizadores como aproximación a la onda sinusoidal, más difícil de generar.

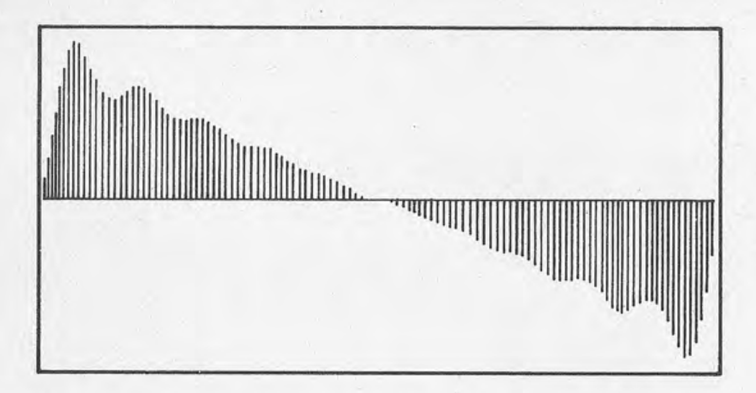


Figura 3.—Onda en forma de "diente de sierra" que aparece en numerosos instrumentos (casi todos los de cuerda y de metal). La onda del dibujo ha sido obtenida al sumar todos los armónicos hasta el décimo con pesos decrecientes.

la variación de intensidad a lo largo de la duración del sonido. Cuando pulsamos una tecla en un piano el sonido no alcanza en seguida la intensidad máxima para mantenerse allí hasta que la soltamos (como puede ocurrir, en cambio, en un órgano electrónico sencillo sin control de la envolvente). Lo que ocurre, en cambio, es un período (muy breve) de aumento del volumen, tras el cual la intensidad disminuye lentamente. En el caso de un instrumento de viento se produce una fase de rápido crecimiento hasta alcanzar un pico, seguida de una disminución hasta un nivel más

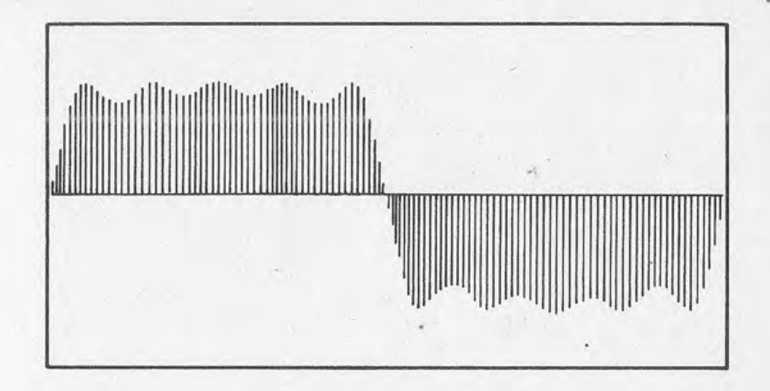


Figura 4.—Onda "cuadrada", resultado de una presencia preponderante de los armónicos impares (en el dibujo se han utilizado los armónicos impares hasta el noveno, con pesos decrecientes). Es típica de instrumentos de viento de madera (clarinete, por ejemplo) o de órganos (no electrónicos).

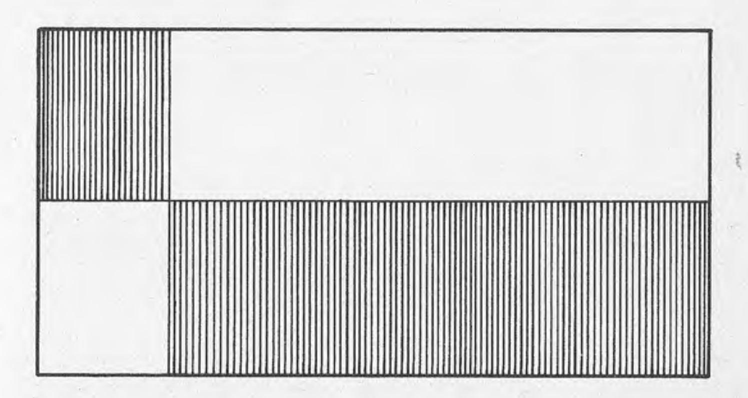


Figura 5.—Onda rectangular ("pulse wave"). A diferencia de la onda de la figura 4, ésta no ha sido obtenida por síntesis aditiva, sino que es el producto típico de un aparato electrónico. Resulta fácil de sintetizar, ya que basta con alternar los dos estados (alto y bajo). Al cambiar la anchura correspondiente de los dos estados se obtienen sonidos bastante variados. La onda del dibujo es una "20% pulse wave", que quiere decir que la anchura del estado alto es un 20% del total.

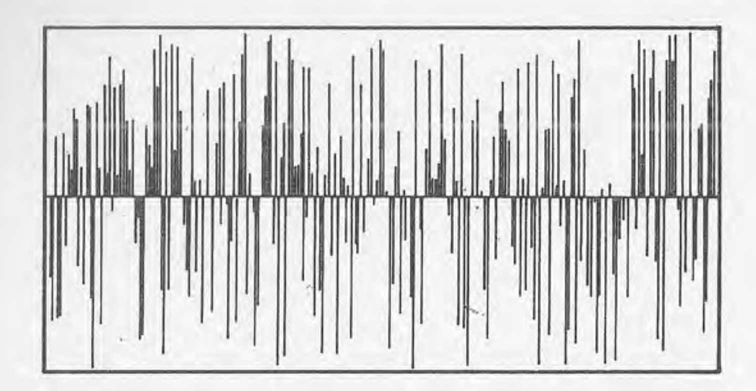


Figura 6.—"Ruido". Esta onda ha sido obtenida al generar casualmente amplitudes sucesivas. El resultado puede servir para efectos especiales (truenos, rayos, explosiones) para, por ejemplo, algún video-juego.

o menos estable en la duración de la nota, después del cual el sonido se apaga con bastante velocidad.

Vemos en este último ejemplo las 4 fases con las que se describe tradicionalmente la envolvente de un sonido: el ATAQUE (attack), primera fase de crecimiento; el DECAIMIENTO (decay), donde el sonido decrece hasta el nivel de SOSTENIMIENTO o MANTENIMIENTO (sustain) y la última fase, la RELAJACION (release), en la cual el sonido baja desde el nivel anterior hasta apagarse (Fig. 7). La duración de estas cuatro fases (no siempre están todas ellas presentes) y las intensidades del pico y del nivel de sostenimiento son las informaciones que permiten describir la envolvente (este recorrido se llama a menudo ADSR por las iniciales de las cuatro fases). Pero tales informaciones a veces no son suficientes: podemos precisar la forma de la pendiente de la curva (puede ser de tipo lineal, logarítmica u otra distinta) y tampoco tiene por qué haber sólo estas 4 fases en la descripción de la envolvente.

Al observar las curvas de las figuras 1 a 7 resulta fácil confundir la forma de la onda con su envolvente. Es verdad que en todos los casos las curvas describen de alguna manera las variaciones de amplitud del sonido en función del tiempo, pero ¡cuidado!: en el caso de la forma de onda tal amplitud varía en el espacio de un período, que dura, como máximo, alguna centésima de segundo, así que al oído "no le da tiempo" a oír la variación de volumen como tal, pero sí reconoce la frecuencia y el timbre. En el caso de la envolvente la duración es la de una nota, del orden de un segundo; es, por tanto, suficiente como para poder captar la variación de volumen.

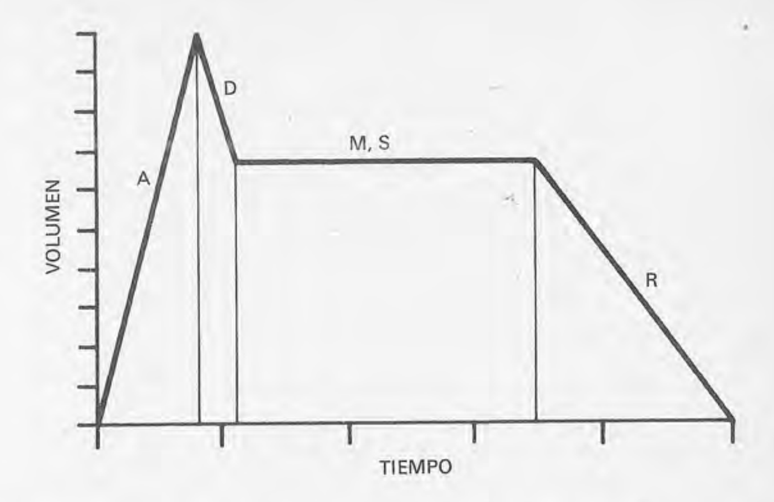


Figura 7.—Cuatro fases que describen la envolvente de un sonido: ataque, decaimiento, mantenimiento o sostenimiento y relajación.

Todo cuanto hemos dicho hasta ahora concierne a la identificación de un único sonido musical, por ejemplo, una nota producida por un solo instrumento. Al escuchar varias notas a la vez, producidas por uno o varios instrumentos, tenemos un cierto número de sonidos, cada uno de los cuales está caracterizado por una frecuencia, un volumen y un timbre. Nuestro cerebro "decodifica" la información global y reconoce los distintos componentes. Si queremos que un sistema electrónico posea la misma riqueza, tendrá que contar con varios osciladores cuyos parámetros puedan ser fijados independientemente. Son las distintas "voces" de nuestro sintetizador.

Tras esta breve introducción ya sabemos sobre qué parámetros actuar si queremos obtener sonidos de un microordenador. Podría dar la impresión de que es fácil reproducir el sonido de cualquier instrumento con sólo elegir detenidamente la envolvente y la forma de onda, pero la realidad es algo más compleja. En primer lugar, la forma de la onda varía normalmente para un mismo instrumento al tocar una nota aguda o una grave. En teoría necesitaríamos una forma de onda particular para cada nota... Y no sólo eso: generalmente hay una variación notable en la forma de la onda en el período de ataque y en el de sostenimiento (por ejemplo) durante la ejecución de una misma nota. No resulta im-

posible tener en cuenta todos estos factores, pero puede llegarse al límite práctico de las posibilidades de un microordenador.

Por otra parte el timbre de un instrumento depende también de la ejecución: un buen músico puede obtener de un instrumento una riqueza de matices difíciles de reproducir incluso con los medios más sofisticados. Aunque, bien mirado, esto en realidad es positivo: por suerte aún queda en el hombre una dosis de imprevisibilidad que sigue haciéndole indispensable...

Reproducir el sonido de instrumentos que ya conocemos puede resultar una labor apasionante y de gran ayuda para un músico profesional, pero seguramente no es el objetivo más interesante en el campo de la música computerizada: el gran mérito de un ordenador es el poder crear sonidos nuevos; más que transcribir a mano una partitura e interpretarla de una manera automática, es interesante crear nuevas melodías o "armonías" desconocidas hasta ahora.

Con demasiada frecuencia se hace uso del ordenador, tanto en música como en gráficos u otros campos, para imitar la actividad humana, a menudo con resultados que desilusionan, ya que no se tienen en cuenta las limitaciones del aparato desperdiciando al tiempo una parte importante de su potencialidad. Esperamos que este libro sirva para que algún lector trate de aprovechar tales posibilidades, siempre que no se desanime tras la primera experiencia (siempre ofrece un resultado inferior al deseado). Se requerirá un cierto esfuerzo, pero al final los resultados confirmarán que valió la pena.

Algo más sobre la forma de onda

Ya hemos visto que el timbre de un instrumento depende en gran medida de la forma de onda asociada al sonido emitido. Pero, ¿a qué se deben las diferencias entre las distintas formas de onda? Las ondas producidas por un sistema físico vibrando a una cierta frecuencia poseen todas, en teoría, la misma forma sinusoidal. Deberíamos tener entonces ondas sinusoidales para cada sonido musical y no existiría la riqueza de timbres de todos los instrumentos construidos por el hombre.

Por suerte esto no es así y la explicación es bien sencilla. Cuando tocamos un LA central, por ejemplo, el instrumento no produce sólo una onda sonora de frecuencia 440 Hz, sino también otras vibraciones secundarias cuyas frecuencias son múltiplos de la de base (llamada fundamental). Tendremos entonces vibraciones de frecuencia 880 Hz, 1.320 Hz, 1.760 Hz... Estas se llaman armónicos y es su presencia la que modifica la forma de onda y da al sonido musical toda su riqueza.

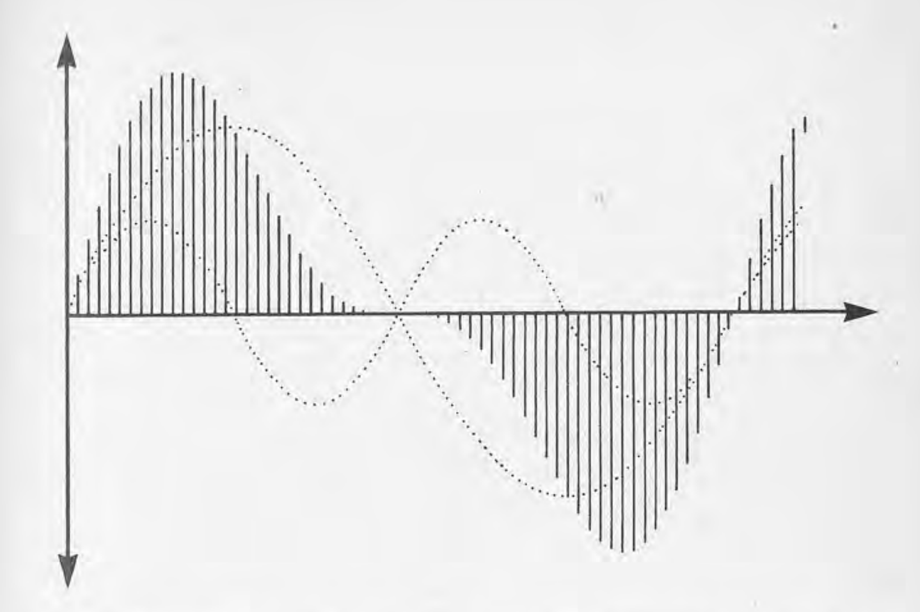


Figura 8.—Ejemplo de síntesis aditiva. Aquí tenemos los dos primeros armónicos (al segundo le hemos dado una amplitud máxima mitad de la del fundamental). La onda resultante (sombreada) tiene ya un aspecto más parecido al "diente de sierra".

La frecuencia fundamental es el primer armónico y corresponde al LA de nuestro ejemplo. El segundo armónico tiene una frecuencia doble del primero y corresponde por tanto a otro LA situado una octava por encima de la fundamental. El tercer armónico tiene una frecuencia triple de la fundamental o, lo que es igual, 3/2 de la frecuencia del segundo armónico. Esta relación de frecuencias corresponde a un intervalo de quinta mayor y la nota correspondiente en este caso sería "casi" un MI (el motivo de ese "casi" se halla en el modo de construcción de la escala utilizada en Occidente y sería largo de contar aquí). El cuarto armónico, de frecuencia cuádruple de la fundamental, se halla dos octavas por encima de ésta (en nuestro caso es un tercer LA) y así todos los demás. Esto que estamos explicando no es válido para una gran parte de los instrumentos de percusión, cuyo timbre es un tanto particular.

La existencia de estos armónicos se justifica por el estudio de las masas vibrantes (cuerdas, tubos de aire, etc.) y puede literalmente "tocarse con el dedo" con un simple experimento: pulsemos la cuerda de una guitarra; la veremos vibrar y oiremos el so-

nido que emite. Mientras vibra apoyemos ligeramente un dedo en la cuerda exactamente en la mitad: oiremos la misma nota, pero más débil y desplazada una octava hacia los agudos. Si la soltamos a tiempo podremos volver a escuchar el sonido original y con un poco de esfuerzo descubrir incluso en este sonido la componente de frecuencia doble que habíamos aislado segundos antes. Hemos resaltado de esta manera la presencia del segundo armónico al lado del fundamental.

Podemos volver a hacer el mismo experimento (aunque el resultado será esta vez menos evidente) tocando la cuerda mientras vibra a un tercio de su longitud para captar el tercer armónico y posteriormente, si se desea, el cuarto, etc. Este fenómeno tiene una fácil explicación: para una tensión fija la frecuencia de vibración de una cuerda es proporcional al inverso de su longitud. En el movimiento vibratorio de la cuerda se producen unos "nodos" en la mitad, en los tercios, en los cuartos..., de la cuerda y a la frecuencia fundamental se superponen sus múltiplos, que son precisamente las frecuencias de los armónicos.

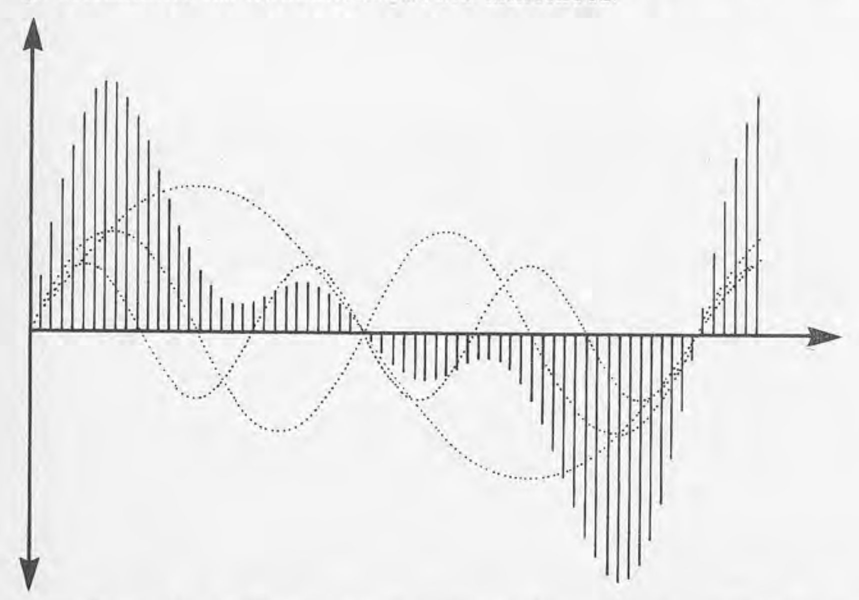


Figura 9.—Otro ejemplo de síntesis aditiva. Hemos puesto los tres primeros armónicos con pesos respectivos 1, 3/4 y 1/2 (la amplitud máxima del tercer armónico es, por tanto, la mitad de la del fundamental). Tanto aquí como en la figura 8 los armónicos están en fase (parten de cero en igual dirección a la vez en el mismo instante). Con ondas desfasadas se obtiene una onda resultante distinta (y un resultado ligeramente distinto para el oído).

La frecuencia de los armónicos va aumentando según el orden de éstos y a un cierto punto la de los armónicos más altos se sale del intervalo que el oído humano puede percibir. El número de los armónicos significativos ("audibles") puede considerarse limitado a una veintena e incluso menos.

La construcción de la onda definitiva a partir de los distintos armónicos puede ilustrarse gráficamente trazando las curvas correspondientes a cada uno de los armónicos sobre un mismo gráfico. Serán todas sinusoides, cuyo período es un divisor entero del período de la fundamental. Al hacer la suma algebraica de las intensidades (es decir, de las ordenadas) en cada instante se obtiene la intensidad resultante en cada punto y puede trazarse la forma de onda final (Figs. 8 y 9).

La forma de onda correspondiente a los instrumentos clásicos se asemeja a menudo a las de las figuras 3 ó 4, de aquí el nombre de ondas "de diente de sierra" o de "onda cuadrada".

El procedimiento que hemos descrito se llama de síntesis aditiva y permite construir la forma de onda de un sonido si se conoce el número de los armónicos y sus pesos respectivos.

La operación inversa, es decir, identificar los armónicos presentes en un sonido, es posible con unos aparatos adecuados y se llama análisis armónico. Con ciertos equipos (el Macintosh o el Music System para Apple II) se puede construir una forma de onda cualquiera sin utilizar una descomposición en armónicos. De todas formas, el análisis armónico juega un papel fundamental en todos los casos. El famoso teorema de Fourier dice que todo fenómeno periódico puede ser definido como la suma (finita o no) de ondas sinusoidales (ha sido demostrado más recientemente que esto es posible también con otras ondas elementales).

CAPITULO II

¿COMO SE LAS ARREGLA EL ORDENADOR PARA EMITIR SONIDOS?



Tras conocer las características de un sonido musical surge, por lógica, otra pregunta: ¿de qué manera logra el ordenador emitir tales sonidos? El tema de la síntesis sonora es bastante amplio, de manera que nos conformaremos con dar tan sólo unas nociones básicas.

Un generador sonoro debe ser capaz de producir una señal eléctrica cuyas variaciones en el tiempo se correspondan con las de un sonido musical en la forma indicada en el capítulo anterior. Esta señal se transmite a un sistema de amplificación y posteriormente a unos altavoces (o auriculares) y... ¡eso es todo!

No resulta muy difícil realizar un circuito eléctrico oscilatorio del que sea posible controlar la frecuencia: esto puede ser considerado como el corazón de un sintetizador analógico.

Durante muchos años los generadores sonoros de los sintetizadores han sido esencialmente de este tipo, teniendo como fuente de la señal uno o varios VCO (Voltage Controlled Oscillator: Oscilador controlado por tensión). Un límite inmediato del dispositivo afecta a la forma de onda. Resulta fácil producir ondas sinusoidales (por las propiedades de los circuitos eléctricos) o rectangulares (que corresponden a la alternancia de dos estados), pero no formas de onda más sofisticadas.

Para disponer de una elección más amplia hay fundamentalmente dos técnicas: la síntesis aditiva y la síntesis substractiva.

El principio de la síntesis aditiva es elemental: como ya sabemos, una señal periódica puede considerarse como la suma de distintas señales sinusoidales de frecuencias múltiples de la fundamental. Bastará con asignar un oscilador (cuyo volumen pueda ser regulado) a cada armónico para que la señal global presente

las características deseadas. El principal inconveniente es obvio: será necesario tener al menos una docena de osciladores para cada voz, motivo que limita en la práctica el uso de la técnica aditiva a los laboratorios de investigación.

La síntesis substractiva sigue un camino inverso: el oscilador produce una onda rica en armónicos (onda tipo diente de sierra o rectangular); estos armónicos serán luego "suprimidos" en parte al utilizar filtros VCF (Voltage Controlled Filters: Filtros controlados por tensión), circuitos que modifican la señal de una manera particular. Un filtro paso-alto "suprime" las frecuencias inferiores a otra elegida previamente (la frecuencia de corte); un filtro paso-bajo actúa al revés y un paso-banda deja pasar sólo las frecuencias que están cerca de la frecuencia de corte.

Con la síntesis substractiva se pierden algunas posibilidades en relación con la aditiva, pero a cambio resulta más fácil construir un timbre "de oído". Naturalmente, un sintetizador corriente ofrece otros dispositivos para obtener una gran variedad de sonidos. Tenemos así un generador de envolvente, el llamado VCA (Voltage Controlled Amplifier: Amplificador controlado por tensión). Existe a menudo la posibilidad de controlar con un VCA también la acción del filtro o del LFO (Low Frequency Oscillator: Oscilador de baja frecuencia) para obtener así un timbre variable durante la emisión de la nota para una mayor semejanza con los sonidos naturales.

Encontramos también el ya citado LFO, un oscilador que opera en una gama de frecuencias inferior a la de los sonidos audibles (como dice la palabra en sí). El LFO produce efectos de vibración (cuando controla la frecuencia del sonido) o de trémulo (cuando lo que controla es el volumen)..

Las cosas se complican algo más si a estos elementos añadimos controles secundarios y si pensamos que a menudo tenemos para cada voz más de un VCO con sus correspondientes LFO, VCA, etc. El crear un nuevo timbre (o la mera reconstrucción de un timbre en particular) requiere mucho tiempo y un gran derroche de energía.

Es entonces cuando vienen en nuestra ayuda las técnicas digitales: el valor de un control, la posición de un regulador pueden estar representados por un número elegido en una escala adecuada. No sólo los circuitos de control se hacen más pequeños, precisos y fiables, sino que es posible además memorizar y solicitar con rapidez un timbre específico (que se halle en cinta, disquete o memoria permanente de otro tipo). Nacen así los llamados sintetizadores híbridos (generadores sonoros analógicos, pero controlados digitalmente) que ocupan hoy en día una gran parte del mercado y que siguen siendo los favoritos de quienes prefieren la suavidad del sonido analógico.

Síntesis digital

Llegamos así a la última etapa de la evolución: la sustitución del oscilador analógico por un generador digital DCO (Digital Controlled Oscilator: Oscilador controlado digitalmente). Basta con tener claro que es posible representar la forma de onda en un período como una sucesión de números (los sucesivos valores de la amplitud). El DCO sacará entonces estos valores a un DAC (Digital-Analogic Converter: Convertidor digital-analógico) para obtener la señal eléctrica que ha de ser amplificada.

Hay distintas técnicas de síntesis digital: pueden utilizarse formas de onda fijas, memorizadas en ROM como tablas de números o construidas en tiempo real o bien pueden reservarse una zona de RAM para tablas de formas de onda definidas por el usuario.

Para construir estas tablas existen instrumentos que realizan patrones de sonidos reales. Estos dispositivos analizan la onda sonora a intervalos pequeñísimos (fracciones de milisegundos) y memorizan los valores hallados. El DCO utiliza estos valores para reproducir el sonido. Esta técnica puede ir aún más lejos: en lugar de repetir en cada período la misma forma de onda puede tomarse el patrón de un sonido que tenga mayor duración (incluso de varios segundos) y memorizar así las variaciones de forma de onda, intensidad, etc. Se hace entonces posible "interpretar" con un simple teclado los instrumentos más extraños o, si se prefiere, reproducir el viento, la lluvia o el ladrido de un perro.

Una técnica de síntesis muy especial es la modulación en frecuencia (FM). No es fácil de explicar en dos palabras, pero lo intentaremos... Se requieren dos osciladores (generalmente de tipo sinusoidal). La señal producida por el primer oscilador (la "portadora") es modulada por el segundo (la "moduladora"). ¿En qué consiste esta modulación? Volviendo a las tablas que representan la forma de onda podemos decir que en lugar de leer la tabla secuencialmente, el generador lee en cada instante en la tabla del primer oscilador el valor que se halla en la dirección indicada por el valor leído en la otra tabla (¡resulta más sencillo de lo que parece!); matemáticamente la señal resultante es la composición de dos funciones de tipo sinusoidal.

Este sencillo dispositivo produce formas de ondas muy variadas al ir modificando la frecuencia y el peso de los dos osciladores. Si cada oscilador dispone de un control propio de la envolvente obtendremos con poco gasto un sonido que varía durante la emisión de una nota no sólo de volumen, sino también de timbre, con efectos muy realistas o extraños.

En la práctica los sintetizadores FM utilizan a menudo más de dos osciladores (operadores) por cada voz, combinándolos en distintas configuraciones (llamadas algoritmos), disponiendo así de

una riqueza notable en cuanto a timbre se refiere. El punto negativo de la síntesis FM es la dificultad para producir el timbre deseado actuando sobre los parámetros, no intuitivos, del generador.

Comercialmente, la síntesis FM ha sido popularizada por los sintetizadores de la serie DX producidos por la empresa YAMAHA, artífices de una pequeña revolución en el mundo de los instrumentos musicales. Esta misma técnica aparece en el ordenador CX5M de esa firma.

Hoy en día la evolución de la música está cada vez más ligada a los avances en informática y en técnicas digitales, tanto a nivel de investigación como al de la música de consumo. En esta última categoría se van multiplicando los aparatos dotados, igual que los ordenadores, de microprocesadores, memoria... Citaremos entre todos ellos el secuenciador, capaz de memorizar una pieza musical tocada en directo con todos los parámetros de la interpretación.

El MIDI

Un nuevo empuje hacia la compenetración entre informática y música se produjo hace tres años con la adopción por parte de todos los constructores de instrumentos musicales del estándar MIDI.

Este protocolo cubre todos los aspectos (tanto de hard como de soft) de la comunicación entre instrumentos electrónicos, secuenciadores e incluso ordenadores. De manera que todo aparato dotado de una salida y una entrada MIDI puede ser conectado a otros dispositivos. Un teclado puede entonces manejar simultáneamente varios instrumentos y enviar o recibir señales a (o de) un secuenciador. Y lo que nos concierne aún más es que un ordenador dotado de su correspondiente interface puede memorizar una secuencia y procesarla, distribuir órdenes a 16 instrumentos, etc. Se ha abierto, por tanto, un nuevo mundo en la informática musical en el que hasta los ordenadores poco dotados musicalmente (tipo Apple, por ejemplo) hallarán nuevas vías.

Limitaciones

Esta exposición acerca del cada vez más sofisticado material musical en circulación nos servirá también para mencionar las limitaciones de los "instrumentos" presentes en un ordenador.

Para empezar, muchos ordenadores no disponen de un verdadero generador sonoro: un simple circuito activa el pequeño altavoz interno por orden del microprocesador central. Cuando nos encontramos frente a un procesador dedicado al sonido suele disponer de un reducido número de osciladores (de 2 a 4). A menudo no hay elección posible de la forma de onda y hay sólo un reducido control de la envolvente. A estas diferencias cuantitativas se añaden otras cualitativas: la relación señal-ruido, la frecuencia patrón y otros no están al nivel de un verdadero sintetizador.

¿Qué queremos decir con todo esto? Tan sólo que no hay que hacerse ilusiones de obtener resultados equivalentes a los producidos por aparatos profesionales que pueden costar millones (sin olvidar tampoco que nuestros modestos ordenadores no se dedican sólo a la música...). Lo que sí podemos comparar es la posibilidad de acercamiento a la informática musical que había ayer (es decir, el vacío total) y la que tenemos hoy. Hace unos pocos años era un sueño pensar en tener por poco dinero un aparato como los de ahora; el hecho de saber que nuestros "instrumentos" son limitados no nos impedirá tratar de sacar de ellos el mejor partido posible.

No podremos imitar todo aquello que escuchamos por la radio, pero sí crear algo nuevo y original. Por otra parte, seguramente no tendremos que esperar mucho tiempo para asistir a un nuevo salto de calidad en la producción musical de los microordenadores, aunque ahora mismo, con el Yamaha CX5M y alguna tarjeta para el Apple, encontramos ya en el mundo de la microinformática material musical con un nivel decididamente profesional.

CAPITULO III

BREVE PANORAMICA DE LOS ORDENADORES MUSICALES

Vamos a examinar ahora qué nos ofrece el mercado en cuanto a ordenadores con vocación musical se refiere. Este capítulo será de gran ayuda, naturalmente, para aquellos lectores interesados en una adquisición de este tipo, pero también para quienes deseen conocer algo más acerca de las posibilidades aún sin explorar de su propio ordenador.

El mercado de la microinformática está en evolución constante, de manera que resulta difícil pretender dar una fotografía completa de la situación, si bien hemos procurado incluir lo fundamental.

Hemos hecho una distinción en tres grandes categorías de ordenadores según el número de voces disponibles, aunque podríamos incluir una cuarta categoría: la de los ordenadores con "voz cero", es decir, sin posibilidad de producir sonidos. Pertenecen a esta categoría numerosos equipos exclusivamente profesionales y algún micro doméstico "del pasado" como los Sinclairs ZX-80 y ZX-81 o el Tandy TRS-80, pero hoy en día ninguna empresa se atrevería a presentar un ordenador, a menos que sea de exclusiva dedicación profesional, desprovisto de posibilidades sonoras.

De cada ordenador tomado en consideración analizaremos dos cosas: las características esenciales del generador sonoro y la capacidad de control de éste mediante el BASIC residente.

Ordenadores con generador monofónico

El generador sonoro es más bien reducido: suelen tener un pequeño altavoz cuya vibración está provocada por impulsos que provienen directamente (o casi) del microprocesador.

Como es lógico, una construcción de este tipo presenta bastantes limitaciones: el único control disponible (además de la duración del sonido) es el de la frecuencia. La forma de la onda es fija (rectangular en la mayoría de los casos) y por regla general no se puede modificar el volumen sonoro por medio del software, lo que impide el control de la envolvente. Resulta, por tanto, bastante pobre si queremos producir algo más que una simple melodía monofónica.

Existe además otra limitación importante: mientras el microprocesador está produciendo sonidos no podrá ocuparse de otra cosa y será prácticamente imposible superponer una melodía a la ejecución de otro programa (bueno, esto no es del todo cierto, pero se requiere una habilidad en lenguaje máquina que no está al alcance de un usuario con conocimientos medios).

En resumen: aquellos que deseen comprarse un ordenador "musical" harán mejor en pasar al siguiente párrafo, aunque esto no quiere decir que no puede hacerse absolutamente nada con estos equipos, ¡de hecho los programas que aparecen en los siguientes capítulos dan buena cuenta de ello! Además, dentro de esta categoría están algunos de los ordenadores más difundidos en el mercado, motivo por el cual habrá a su disposición un software abundante que puede de alguna manera suplir las limitadas posibilidades del material. También existen importantes posibilidades de expansión hardware (tarjetas musicales, interfaces MIDI, etc.) que pueden convencernos para fijar la elección precisamente en alguno de estos ordenadores.

SINCLAIR ZX-SPECTRUM

En el interior del Spectrum se encuentra un pequeño altavoz y sus reducidas posibilidades musicales (que, por otra parte, están muy bien aprovechadas en relación con los juegos) no han impedido al sucesor del ZX-80 alcanzar el éxito.

El BASIC dispone de una única instrucción sonora (BEEP), completada por dos parámetros para especificar la duración y frecuencia del sonido. Un valor entero dado a este último parámetro indica una nota de nuestra escala; otros valores permitirán producir sonidos de frecuencias intermedias. Queremos hacer notar la disponibilidad de interfaces MIDI que le dan un carácter decididamente más musical.

El Spectrum 128 supone un desarrollo notable en el campo musical respecto del modelo anterior: aun con una sola voz, dispone de un interface MIDI incorporado.

SINCLAIR QL

También el QL dispone de una sola voz que puede ser controlada únicamente en cuanto a frecuencia se refiere. Pero el sonido no está gestionado por el microprocesador central (68008), sino por otro procesador: el 8049, que se encarga también de la gestión del teclado y de alguna operación de i/o, lo que nos ofrece alguna posibilidad suplementaria. La emisión sonora puede realizarse al mismo tiempo que la ejecución de un programa. Además, la instrucción BEEP del BASIC permite algún efecto especial: es posible, por ejemplo, especificar como parámetro una segunda frecuencia y la velocidad de paso de una a otra para obtener efectos de variación progresiva del tono.

APPLE II

Cuando el Apple II nació, el hecho de utilizar un procesador dedicado a la síntesis sonora era todavía un sueño, no resultan sorprendentes, por tanto, las limitadas posibilidades de los ordenadores de la familia II (II+, "e" y "c").

Lo que sí sorprende es la falta total de instrucciones sonoras en el BASIC Applesoft. Para activar o desactivar el altavoz hay que recurrir a PEEKs de la dirección $C030 (49200 decimal) y para producir una simple tonadilla tendremos que utilizar una subrutina en lenguaje máquina. Por suerte, la difusión del Apple no hace difícil la búsqueda de programas musicales, algunos de los cuales son excelentes (hay incluso uno que simula una emisión sonora polifónica) y aprovechan al máximo la escasa potencialidad del aparato.

Además están las ranuras (slots). No sabemos qué se le pasó por la mente a Steve Wozniak cuando decidió dotar al Apple II de ocho ranuras de expansión (en una época en que no había ni con qué rellenar una), pero el hecho es que sólo las tarjetas musicales producidas para este ordenador son suficientes para saturarlas. Precisamente entre éstas se hallan algunos de los más interesantes generadores sonoros existentes en el mundo de la microinformática.

La tarjeta musical más extendida en la actualidad es la Mockinboard, producida en América por la empresa Sweet Micro. Son numerosos los programas (sobre todo juegos) previstos en su versión estándar para hacer uso de esta tarjeta. Existe en varias versiones, una de las cuales permite también la síntesis vocal. La parte musical se sirve de un generador estereofónico a tres voces con control de la envolvente (basado en el AY-8910 de Ge-

neral Instruments). Por desgracia, esta tarjeta no ha llegado, al menos oficialmente, a España en el momento de escribir este libro.

Otra expansión sonora para el Apple es el Music System (de la firma Mountain Computer) que, si bien no es de lo más reciente en cuanto a concepción, posee unas características bastante interesantes. Dispone de 16 osciladores independientes. Se puede escoger para cada oscilador la frecuencia (sobre 16 bits), el volumen (sobre 8 bits) y la forma de onda. Esta forma de onda no está preestablecida: se puede construir en memoria en forma de una tabla de 256 valores de 8 bits que es consultada en cada período para controlar el oscilador. En el campo de la microinformática sólo el Macintosh ofrece algo parecido.

La capacidad de programación del Music System ofrece como contrapartida que no es fácil de utilizar: por ejemplo, la elección de la envolvente es libre, pero debe ser controlada por una subrutina en lenguaje máquina que modifique el volumen en función del tiempo.

El software proporcionado permite componer piezas musicales polifónicas en el pentagrama dibujado en la pantalla y crear imitaciones de instrumentos reales o invenciones de sonoridades nuevas utilizables en el primer programa.

Otras empresas han ideado teclados vía interface con el Music System (Alpha Syntauri y Soundchaser de la firma Passport Design), acompañadas por un software adecuado, que transforman el Apple en un sintetizador de nivel semiprofesional. Hay que tener en cuenta los límites de la tarjeta, que pueden resumirse en la resolución no precisamente fantástica, de los distintos parámetros fundamentales, pero queda aún un amplio espacio que puede ser aprovechado tanto por el músico como por el aficionado.

IBM-PC y compatibles

Aquellos usuarios que compran un IBM-PC o un compatible no suelen hacerlo precisamente para dedicarse a experimentaciones musicales, pero hay que reconocer que dentro de esta categoría de ordenadores profesionales, en la que están sólidamente asentados, los PCs ofrecen algo más.

Naturalmente poseen un pequeño altavoz que produce un sonido del que sólo se puede controlar la frecuencia con la instrucción SOUND f,d ("f" es la frecuencia, "d" la duración), pero el BASIC Microsoft no se conforma con esto y presenta otras instrucciones menos corrientes. La instrucción PLAY acepta como parámetro una cadena en la que es posible codificar una corta melodía. El metalenguaje utilizado es muy parecido al utilizado en la

homónima instrucción del MSX-BASIC (cosa que no es una coincidencia: recordemos el origen común de ambos).

Otra ventaja es que la potencia de cálculo permite que el ordenador "suene" durante la ejecución de un programa en BASIC. La gestión de esta posibilidad está facilitada por las instrucciones ON PLAY (n), GOSUB XX y PLAY ON (OFF, STOP) que permiten rellenar el buffer musical cuando tiende a vaciarse.

Si sumamos las posibilidades de interconexión del IBM-PC y de la mayor parte de sus "hermanos" con la potencia disponible, estas máquinas pueden imponerse como una de las buenas elecciones para un músico "serio".

SHARP MZ-80 y MZ-700

La familia Sharp de micros domésticos dispone de un sabio generador sonoro de una voz gestionado por un chip distinto del microprocesador, cosa que ofrece alguna ventaja adicional.

En BASIC la instrucción MUSIC permite escoger, como suele ocurrir en esta categoría de ordenadores, el tono y la duración del sonido. TEMPO modifica la velocidad de ejecución.

Ya veremos que el MZ-800 está algo mejor dotado.

DRAGON

Tan sólo dos palabras acerca de este micro doméstico. El sonido es emitido por el televisor conectado al ordenador por medio de la salida RF. Es posible controlar la frecuencia y la duración con la instrucción SOUND. Un punto a su favor es la existencia de la instrucción PLAY, con la cual se puede programar una pieza musical utilizando un metalenguaje parecido al del BASIC MSX.

OLIVETTI M10 (y TANDY-100)

También este portátil Olivetti posee un pequeño altavoz y la instrucción SOUND (en BASIC) que emite un sonido con la frecuencia y duración requerida; con él no podremos componer sinfonías, pero sí llegaremos a impresionar a más de una persona...

EPSON HX-20 y PX-8

Algo parecido ocurre con los dos portátiles de la empresa japonesa: están provistos de un pequeño altavoz interno y de la instrucción SOUND con elección de la frecuencia y duración de la nota.

Ordenadores con generador polifónico (de dos a cuatro voces)

Tratamos ahora de una categoría que se hace cada vez más amplia y que no cuenta sólo con ordenadores de uso estrictamente doméstico. Dotados de salida sonora polifónica, la gestión del sonido es obra de un microprocesador diferente del central. Esto no sólo ofrece una variedad de efectos mucho más amplia, sino que libera además a la C.P.U. de la labor de generación sonora. De esta manera el ordenador puede emitir música mientras desarrolla otra actividad, que no sufrirá interrupciones o retrasos de manera aparente. El procesador que gestiona el sonido puede tener también otros cometidos dentro del funcionamientos del ordenador, pero lo corriente es que se trate de un integrado especializado en esta función. El de mayor difusión es el 8910 de General Instruments (usado, en particular, en los sistemas MSX); es necesario citar también al famoso SID del Commodore 64.

Generalmente estos procesadores poseen tres (a veces dos) osciladores digitales, posiblemente asociados a uno o varios generadores de ruido blanco. Como es lógico, se puede controlar la frecuencia y el volumen para cada voz. A menudo se halla también incorporado un dispositivo de control de la envolvente que puede ser de dos tipos: elección limitada a un cierto número de envolventes predefinidas, o control más completo de tipo ADSR.

Debemos hacer otra distinción importante entre los generadores con forma de onda fija (normalmente rectangular) y los que ofrecen una elección entre varias formas de onda o, menos corrientemente, permiten que el usuario las construya.

La señal sonora se modula generalmente para que sea emitida por el altavoz del televisor conectado a la salida RF del ordenador. Algunas veces está prevista también una salida de audio separada (si utiliza la instalación de alta fidelidad de su casa obtendrá una gran mejora en los resultados), pero otras tendrá que contentarse con el altavoz incorporado al ordenador. La calidad del sonido producido no alcanza, naturalmente, un nivel profesional, pero al tratar de sacarles el mayor partido posible estos aparatos pueden dar grandes satisfacciones.

COMMODORE 64 y 128

Sin duda alguna el C64 constituye una piedra básica dentro de la breve historia de la microinformática musical. Gracias a él centenares de miles de usuarios han tenido acceso por vez primera a un generador sonoro polifónico con extenso control de la forma de onda y de la envolvente.

Pero como no todo puede ser miel sobre hojuelas este ordenador tiene sus defectos: no hay una sola instrucción en el BASIC dedicada a la conversación con el SID (el procesador sonoro del 64). El único medio a disposición del programador en BASIC es la utilización de una serie innumerable de PEEK y POKE, que no tardan en desanimar a los más deseosos. Por eso la gran mayoría de los usuarios del C64 debe contentarse con escuchar los efectos sonoros producidos por los programas comerciales sin lograr obtener el más pequeño sonido de su propio ordenador.

Vamos a comentar unos pocos detalles acerca del SID, chip "custom" presente en el 64 y en el 128. Contiene tres osciladores digitales cuya forma de onda se puede seleccionar individualmente entre onda rectangular, triangular, en diente de sierra y ruido blanco. Para la onda rectangular es posible elegir (sobre 12 bits) la anchura de la parte alta de la onda ("pulse"). La señal producida por cada oscilador pasa a través de un DCA, generador de la envolvente que permite elegir la magnitud de las cuatro fases clásicas en el esquema ADSR (sobre 4 bits cada una). La señal resultante pasa a través de un único filtro para las tres voces. En cuanto al filtro pueden escogerse: el tipo (paso bajo, alto o banda), la frecuencia de corte y las voces que hay que filtrar.

Completan el cuadro otras posibilidades utilizadas para efectos especiales: modulación en anillo (ring modulation), sincronización de una voz con otra, posibilidad de leer la frecuencia y el volumen de una voz (esta capacidad tan poco aprovechada puede ser utilizada, entre otros casos, para elaboraciones en tiempo real).

Tenemos entonces un buen control sobre la forma de la onda (las cuatro formas de onda fundamentales pueden ser combinadas y luego modificadas con una filtración adecuada), un control aceptable de la envolvente y varias posibilidades de efectos especiales; el SID sigue siendo hoy en día el generador sonoro más potente montado de fábrica en un microordenador si excluimos, naturalmente, el del Yamaha CX5M y (ya dentro de otra categoría de precios) el del Macintosh.

Como es lógico, el sonido producido no presenta niveles de calidad profesional: además del número limitado de voces y de otros controles la señal sonora pierde en la definición y en la relación señal-ruido, por ejemplo. Pero el C64 no ha pretendido nunca rivalizar con productos de un precio bien distinto (y no hay

que olvidar que éste no es sólo un instrumento musical, sino también un ordenador...)

Si el proceso de programación del C64 es un tanto complejo podemos consolarnos un poco con el numeroso software de calidad disponible para aplicaciones musicales. El tan difundido Basic Simon suple de alguna manera la falta de instrucciones especializadas del lenguaje residente. Encontramos también bastantes accesorios hardware, interfaces MIDI y teclados musicales.

Pero no podemos terminar sin mencionar que el hermano mayor del 64, el Commodore 128, posee las mismas capacidades musicales y, además, un BASIC residente mucho más completo con instrucciones dedicadas a la gestión del sonido.

COMMODORE 16 y PLUS 4

Estos productos de la firma Commodore no han tenido el mismo éxito en el mercado que sus hermanos VIC-20 y C64. En el campo musical está claro que los C16 y Plus 4 no están a la altura del C64: desaparecido el SID, el sonido está gestionado por un integrado encargado también, entre otras cosas, de los gráficos. Dispone de dos canales con forma de onda fija, con elección de la frecuencia (10 bits) para cada voz y del volumen (4 bits) de la señal global. No posee ni filtro, ni control de la envolvente ni efectos especiales.

Aunque en contrapartida estos nuevos Commodore ofrecen un fácil control del sonido desde el BASIC. La instrucción SOUND acepta tres parámetros para la elección del canal y de las frecuencias y duraciones ecorrespondientes. El volumen global obedece a la instrucción VOL.

VIC-20

No podemos terminar de hablar de los productos Commodore sin mencionar el VIC-20.

Su capacidad musical es más parecida a la del C16 y Plus 4 que a la del C64. Posee algo más en cuanto a número de voces se refiere (tres y un canal de ruido) y algo menos en cuanto a definición sonora (la frecuencia se define sobre 7 bits en lugar de 10). No hay variación de la forma de onda ni de la envolvente y posee un control del volumen (4 bits) único para las tres voces; además, tampoco posee comandos en BASIC para control de sonido.

TEXAS TI 99/4A

Encontramos en el TI 99 un clásico generador de tres canales (en forma de onda fija), además de un canal de ruido. Es posible establecer por cada canal la frecuencia y el volumen. En BASIC es necesario utilizar la instrucción CALL SOUND, que admite como parámetros la duración global del sonido seguida del volumen y de la frecuencia de cada nota.

MSX

Un elemento del estándar MSX es la presencia del generador de sonidos programable AY-3-8910 de la firma General Instruments. Este procesador dispone de tres canales con forma de onda fija (rectangular) además de un canal de ruido que puede ser mezclado con cada una de las otras tres voces. La frecuencia se define sobre 12 bits (5 bits para el ruido), y la amplitud sobre 4 bits para cada voz.

Hay también un control limitado de la envolvente: existen ocho tipos de envolventes seleccionables, de los que es posible establecer la duración (sobre 16 bits). Hay que decir también que sólo dos o tres de estas envolventes son aptas para la emisión de sonidos musicales; las otras están reservadas para ruidos tales como disparos, rayos y otros que gustan tanto a los aficionados a los vídeo-juegos.

El sonido sale del altavoz del televisor o de un equipo de alta fidelidad al que esté conectado.

Como vemos, las posibilidades musicales de los sistemas MSX son buenas pero no excepcionales. Lo que distingue a estos ordenadores es la potencia de las instrucciones musicales del MSX-BASIC. La instrucción SOUND permite modificar directamente los registros del 8910. La instrucción PLAY pone en marcha un verdadero microlenguaje musical muy extenso y versátil. El parámetro de esta instrucción es una o varias cadenas en la que están codificadas las notas que van a sonar, el volumen, el tiempo, la envolvente, etc.

El mundo musical de los MSX va más allá del 8910: empiezan a verse periféricos tipo interfaces MIDI, teclados o verdaderos sintetizadores como el Toshiba, del que hablaremos más adelante. Trataremos también del ordenador MSX Yamaha, que abre nuevos horizontes en el mundo de la microinformática musical.

Pero la cosa no termina aquí: la evolución del estándar, convertido en MSX2, prevé la posible adaptación a cada ordenador de esta familia de un chip musical FM (de modulación de frecuencia) de 9 voces. El generador debería ser igual al que se halla en

el teclado-sintetizador Toshiba. No estará, por tanto, a la altura del de Yamaha, pero el salto de calidad con relación al estándar actual será notable.

ATARI 130 XE, 800 XL, 800 y 400

El Atari 800 y su hermano el 400 fueron en el 79 los primeros microordenadores dotados de un procesador sonoro especializado. Este procesador "custom", el POKEY (que gestiona también el teclado y la salida serie) vuelve a aparecer en el 130 XE, cosa que, por otra parte, no resulta extraña, ya que (sin contar con la capacidad RAM) se trata prácticamente del mismo aparato. Esto también vale para el 800 XL, modelo intermedio entre los dos.

En su momento, el sonido del Atari era algo excepcional. La avalancha de nuevos productos ha dejado un poco en la sombra a estos modelos, pero las características del Atari se han mantenido a un nivel intermedio. Destaca el número de canales (cuatro) con un control reducido de la forma de onda. Esta es alterada con la modificación del parámetro llamado distorsión del Atari (probablemente se trate de la modificación de la anchura de "pulse" de una onda rectangular). Está algo reducida la extensión sonora (tres octavas). En BASIC hay que utilizar la instrucción SOUND con cuatro parámetros: canal, frecuencia, distorsión y volumen. La emisión sonora sigue adelante hasta la ejecución de una instrucción SOUND canal 0,0,0.

ATARI 520 ST

No se sale de la media de las producciones actuales en lo que se refiere a emisión sonora. El procesador utilizado es el clásico AY-3-8910 con sus tres osciladores (además del generador de ruido blanco) y control de la envolvente en sus cuatro fases ADSR.

Posee una característica que llama la atención: el 520 ST es el único microordenador en circulación (si excluimos el ordenador "especializado" Yamaha CX5M) que está dotado en su configuración estándar de un interface MIDI. Añadimos también la velocidad del microprocesador 68000, la confortable capacidad de memoria y podemos imaginar las posibilidades de archivo y de elaboración en tiempo real a nuestra disposición. El precio es muy competitivo, por lo cual el 520 ST estará entre las elecciones obligadas del músico en vías de "computerización".

AMSTRAD CPC 464, CPC 664, CPC 6128

De reciente aparición en el mercado español, los equipos Amstrad están causando un verdadero furor.

El procesador sonoro es un 8912 de General Instruments, con tres canales de forma de onda fija sobre siete octavas. El Amstrad posee un altavoz interno, pero también una salida estéreo para amplificador externo. El BASIC (muy completo) se sirve de la instrucción SOUND para emitir una frecuencia elegida sobre un canal determinado (la emisión sonora dura hasta la ejecución de la instrucción RELEASE). Encontramos también otras instrucciones interesantes: ENV para construir la envolvente del sonido con elección de la duración y amplitud de sus 5 segmentos, y ENT que ofrece posibilidades análogas para la variación no del volumen, sino de la frecuencia (para diferentes efectos, tipo vibración).

La gestión de las interrupciones del BASIC con la instrucción EVERY XX GOSUB YY facilita el control de la duración de la emisión sonora.

En conjunto, por tanto, no están nada mal...

SHARP MZ-800

La última creación en micros domésticos de Sharp se ha puesto al paso con los tiempos al adoptar un procesador sonoro de tres canales sobre seis octavas, además de un generador de ruido. La forma de onda es fija, pero encontramos un discreto control de la envolvente.

Programas en "cartuchos" permiten componer música sobre el pentagrama dibujado en la pantalla, construir nuevos instrumentos, y otros que están llegando al mercado..

La empresa Yamaha se ha empeñado en hacer las cosas en serio en este mundo de la microinformática, en el que se ha integrado recientemente. El CX5M no ha tenido aún la difusión que se merece; pero si logra usted encontrar uno y escucharlo no se sentirá desilusionado...

Otros ordenadores de prestaciones análogas alcanzarán, antes o después, al CX5M en esta categoría, pero no les será fácil, pues la ventaja adquirida por la firma Yamaha en un campo en el que se siente cómoda es, por el momento, considerable.

ACORN BBC y ELECTRON

Estos micros domésticos, que presentan muchas características interesantes (entre las cuales se incluyen las sonoras) no han

tenido un gran éxito en España debido, en parte, a una política comercial poco agresiva.

El generador sonoro opera sobre tres canales en forma de onda fija más un canal de ruido con posibilidad de control de la envolvente.

El BASIC, de estos ordenadores muy completo, contiene dos instrucciones para el tratamiento musical. La instrucción genérica es SOUND, con cuatro parámetros: canal, amplitud, frecuencia y duración. La instrucción ENVELOPE permite la construcción de la envolvente en los segmentos sucesivos que la componen de forma análoga a la descrita para el Amstrad.

ORIC

He aquí otro ordenador inglés que tampoco ha tenido un gran éxito.

Tan sólo dos palabra sobre las nada desdeñables posibilidades musicales del Oric: tres canales sobre ocho octavas, 7 envolventes estándar para elegir. la programación en BASIC está facilitada por las instrucciones MUSIC, PLAY y SOUND que permiten actuar en distintos niveles sobre la generación sonora.

MACINTOSH

Se ha dicho a menudo que el Macintosh es un maravilloso (y caro) juguete; su capacidad musical no va precisamente en desacuerdo con esta afirmación.

Podemos tranquilamente afirmar que (si excluímos el Yamaha) éste es en la actualidad el único microordenador que posee de serie el generador sonoro más potente.

Tiene, además, una característica única entre los de su clase: la forma de onda de cada uno de los cuatro canales puede ser totalmente definida por el usuario en forma de una tabla de 256 elementos que hay que construir en memoria. La frecuencia de base es de 22 KHz, de manera que es suficiente para reproducir sin alteración los sonidos, si excluimos los de frecuencia muy alta.

Naturalmente es posible controlar el volumen de cada voz. No posee control directo sobre la envolvente, sino que hay que actuar directamente sobre el volumen.

Hay una gran diferencia entre las dos versiones del BASIC Microsoft en cuanto a las instrucciones sonoras se refiere. La versión 1.0 propone una única instrucción BEEP que produce una nota de la que podemos especificar el timbre, la frecuencia y la duración. La versión 2.0 aprovecha mucho más a fondo las capacidades musicales del Macintosh: con la instrucción SOUND es posible emitir sonidos polifónicos y con la instrucción WAVE podemos construir libremente las formas de onda que queramos.

Hay excelentes programas musicales para el Macintosh pero en su conjunto las capacidades musicales de este ordenador no han sido muy aprovechadas, aunque también es cierto que hay muchas más cosas con las que «jugar» en un Macintosh.

COMMODORE AMIGA

Las posibilidades del generador sonoro de este ordenador son bastante buenas. Se pueden programar libremente los cuatro canales sonoros (sobre 9 octavas), creando en memoria la forma de onda y la envolvente que queramos. Cada canal dispone de un DAC (digital - to - analog converter, convertidor analógico-digital) de 8 bits y un control de volumen (sobre 6 bits).

El Amiga dispone de dos salidas sonoras para conectar a un equipo estéreo y, lo que resulta más raro, dispone también de dos canales de entrada por medio de los cuales es posible digitalizar un sonido. A continuación, podremos reproducir fielmente el sonido memorizado (tras una posible elaboración). La frecuencia de base es programable, lo que permite elegir la relación calidad del sonido - ocupación de memoria más favorable.

Microordenadores polifónicos (8 voces y más)

Debido a la diferencia que existe en relación con la categoría precedente nos hemos visto obligados a señalar aparte esta serie de ordenadores, con el peligro de encontrarnos con un solo competidor en el campo.

Para que el CX5M de la firma Yamaha no estuviera tan solo hemos incluido también en este apartado el teclado-sintetizador de Toshiba, compatible MSX, sin olvidar tampoco que el Music System para Apple estaría por derecho en esta categoría. Aunque, en honor a la verdad, debería haber otro contendiente más: cuando Jack Tramiel anunció la renovación de la gama Atari (con el 130XE y el 520ST) recordamos haber oído mencionar un modelo 65XEM. Este último no es sino un normalísimo 65XE dotado de un revolucionario chip musical llamado AMY. Este procesador dispone de 64 osciladores digitales para producir 8 voces independientes. Todavía no hemos logrado conocer la técnica de síntesis sonora, si es de modulación de frecuencia o de formas de onda prefijadas. Lo que sí es seguro es que, según los pocos elegidos que lo han escuchado, la riqueza de timbre que prometía debe

ser notable y la calidad del sonido muy elevada. Más bien deberían serlo, pues desde el día de la presentación no se ha vuelto a oír hablar del AMY ni del 65XEM y empezamos a tememos que éste no aparecerá nunca en el mercado, lo que sería una lástima...

A pesar de que tendremos que contentarnos con un único protagonista dentro de esta serie que estamos presentando, hay que considerar al CX5M mas como el primer representante de un nuevo tipo de ordenadores que como una creación aislada.

Con los avances en la tecnología de los circuitos integrados y en la ciencia de la síntesis musical se hace posible la realización de módulos sintetizadores a nivel profesional que poseen dimensiones y precios cada vez más razonables. Resulta difícil prever la forma de esta evolución: ordenadores de uso general que posean además un sintetizador de alto nivel o instrumentos musicales que integren un verdadero microordenador. Podemos decir con toda seguridad que veremos aparecer una nueva familia de aparatos cada vez más fascinante, de la que el CX5M es ya un brillante miembro.

Sintetizadores TOSHIBA HX-MU 900 MSX

Este modelo no ha sido aún comercializado por lo que tendremos que limitarnos a una presentación incompleta.

La unidad HX-MU 900 integra un teclado musical (4 octavas) mas sintetizador y se conecta a cualquier sistema MSX por medio del slot para cartuchos.

La técnica de síntesis utilizada es la modulación de frecuencia. Hay 9 voces que disponen cada una de 2 operadores, agrupables según 2 algoritmos para generar sonidos sobre 8 octavas. La calidad del sonido (en particular la versatilidad de timbre) se destaca sensiblemente de la de los clásicos generadores sonoros de micros domésticos, aunque no llega al nivel de sintetizadores FM más sofisticados (tipo Yamaha por ejemplo). Es posible escoger entre 65 timbres ya programados en ROM que imitan instrumentos reales (para algunos el resultado es bueno, para otros no resulta muy convincente).

En la versión de base el HX-MU 900 se utiliza como un clásico teclado musical polifónico con alguna sofisticación (tipo ritmos programables). Más adelante debería haber también disponible un software dedicado: un programa de composición musical y un programa de creación de nuevos «instrumentos». Aun no se sabe si será posible programar el sintetizador a nivel de BASIC o de lenguaje máquina. En los próximos meses habrá sin duda alguna información más extensa, aunque podemos afirmar ya que el teclado Toshiba ofrece a los amantes de la música poseedores de un MSX una «quinta marcha».

YAMAHA CX5M

El CX5M presenta un doble aspecto. Por un lado se trata de un clásico ordenador MSX compatible con todo el software y los accesorios previstos para este estándar, por otro es un sintetizador musical cuyas prestaciones superan con mucho todo lo que hemos oído hasta ahora en el mundo de la microinformática: ocho voces independientes, síntesis de sonido en modulación de frecuencia que permite una gran riqueza de timbres (cada voz posee 4 operadores que pueden ser combinados según 8 algoritmos), programabilidad extensa de la envolvente de cada operador que permite un control no sólo sobre la evolución del volumen en la duración de la nota, sino también sobre la del timbre; 48 instrumentos predefinidos en ROM además de los definibles por parte del usuario, secuenciador incorporado, interface MIDI, etc.

Ya hablamos en el capítulo 2 de la modulación de frecuencia. Hay que recordar que ha sido la misma firma Yamaha la que ha difundido comercialmente esta técnica con sus sintetizadores de la serie DX, que siguen estando hoy en día entre los favoritos de muchos músicos. En lo restante, el módulo presente en el CX5M es idéntico al del DX9 o al del reciente DX21.

El resultado es un sonido de calidad decididamente profesional que puede ser disfrutado al máximo con un sistema de amplificación de buen nivel. Los instrumentos proporcionados resultan en conjunto muy convincentes (son muy buenos en particular los instrumentos de viento y de percusión). A aquellas personas que han oído alguna vez el sonido CX5M les resulta muy difícil volver al de otro ordenador...

Vamos a describir rápidamente la puesta en marcha de nuestro «instrumento». El sintetizador está contenido en un módulo extraíble a la izquierda del ordenador. Resultan visibles las salidas de audio, las dos tomas del interface MIDI y la toma para uno de los teclados dedicados. Una vez conectados el teclado y la instalación de amplificación tendremos a nuestra disposición un clásico sintetizador que incorpora un secuenciador monotraza y un generador automático de ritmos y acompañamientos. Podemos leer los distintos parámetros en la pantalla y cambiarlos utilizando el teclado del ordenador.

Si queremos pasar a la programación del sintetizador tendremos alguna desilusión: la firma Yamaha no proporciona actualmen-

te ninguna documentación técnica sobre el sintetizador (aunque parece ser que la situación está cambiando y los programadores tendrán dentro de poco acceso a toda la información necesaria). Está disponible en cartucho el software Music Macro que añade unas treinta palabras nuevas al BASIC para la programación musical. Las instrucciones son numerosas y potentes, pero algunos comandos para acceder a los parámetros de los osciladores y ciertas operaciones del BASIC son demasiado lentas.

	Número de voces	Octavas	Forma de onda	Control de la envolvente
SINCLAIR SPECTRUM	1	10	fija	—
SINCLAIR QL	1	10	fija	—
APPLE II+, IIe, IIc	1	4	fija	—
IBM-PC (e compatibili)	1	7	fija	—
SHARP M2-80 y M2-700	1	3	fija	—
DRAGON	1	5	fija	—
OLIVETTI M10	1	5	fija	—
EPSON HX-20 y PX-8	1	5	fija	—
COMMODORE 64 y 128	3	8	4 formas de onda	Extenso
COMMODORE 16 y PLUS 4	2	6	fija	Sólo control del volumen
COMMODORE VIC 20	3 (+ ruido)	3	fija	Sólo control del volumen
TEXAS TI 99/4A	3 (+ ruido)	4	fija	Sólo control del volumen
MSX	3 (+ ruido)	8	fija	Limitado
ATARI 130 XE, 800 XL	4	3	Variaciones limitadas de la forma de onda	Sólo control del volumen

	Número de voces	Octavas	Forma de onda	Control de la envolvente
ATARI 520 ST	3 (+ ruido)	8	fija	limitado
AMSTRAD CPC 464, 664, 6128	3	7	fija	Extenso
SHARP MZ-800	3	6	fija	Limitado
ACORN BBC y ELECTRON	3 (+ ruido)	5	fija	Extenso
ORIC	3 (+ ruido)	8	fija	Limitado
SEGA SC 3000	3 (+ ruido)	8	fija	Sólo control del volumen
APPLE MACINTOSH	4	12	Completamente programable	Sólo control del volumen
COMMODORE AMIGA	4	9	Completamente programable	Programable
YAMAHA CX5M	8	8	Programable (síntesis FM)	Muy extenso

CAPITULO IV

¡MANOS A LA OBRA!

Después de tanta teoría llegamos por fin a una actividad más excitante: ponernos a trabajar para obtener de nuestros ordenadores sonidos, ruidos y (¿quién sabe?) tal vez inolvidables composiciones musicales.

En los siguientes capítulos encontraremos muchas ideas para poner en práctica y bastantes programas en BASIC. Estos programas tienen una doble utilidad: pueden considerarse programas listos para ser tecleados y obtener resultados inmediatos, pero también esperamos que resulten útiles, sobre todo, como punto de partida para crear producciones más personales y elaboradas. El objetivo de este libro, como ya dijimos, es el de servir de ayuda para emprender los primeros pasos en el amplio mundo de la informática musical.

El hecho de pasar a la práctica no ha sido tarea fácil: es cosa sabida que el BASIC estándar es un concepto totalmente teórico. No sin cierto esfuerzo se puede hallar un mínimo de compatibilidad en las instrucciones de uso general, pero si pasamos a la gestión de la música nos aventuraremos en una verdadera jungla que hace prácticamente imposible la traducción de un programa destinado para un ordenador específico. Y esto no se produce sólo porque las posibilidades musicales varían de un modelo a otro, sino también porque la gestión de capacidades análogas ha sido abordada con una filosofía totalmente particular por cada marca.

Para hacer frente a una situación de este tipo (que es peor incluso que la que se produce en el campo gráfico) hemos tenido que tomar una decisión. Por un lado no era posible dar la versión de cada programa para cada ordenador en estas pocas páginas.

Por otro no podíamos publicar programas sólo, pongamos por caso, para el C64 y provocar la ira razonable de los demás usuarios.

Hemos optado entonces por una vía intermedia: cada programa lleva consigo un cuerpo central, escrito en un BASIC "estándar" que debería funcionar sin problemas (o casi, preferimos añadir...) en todos los ordenadores. Este tronco llama a distintas subrutinas, principalmente para la gestión de las actividades musicales. La numeración de estas subrutinas es la misma en cada programa para evitar repeticiones inútiles.

Esta tentativa comporta, obviamente, algunas limitaciones: para tener programas que funcionen en todos los ordenadores habrá que contentarse con aprovechar las posibilidades comunes a todos ellos. De esta manera el sonido se emite sólo por un canal y los únicos controles afectan a la frecuencia y a la duración de la emisión sonora. Aquellas personas que posean ordenadores con más potencia encontrarán alguna indicación para sacarles mayor provecho.

Por este mismo motivo los programas tendrán que tener una presentación más bien austera, sin gráficos ni florituras. Dejamos como ejercicio para el lector las mejoras del interface de usuario para su ordenador en concreto.

Forma de usar los programas

1) Teclee uno de los programas cuyo listado está publicado en los próximos capítulos.
2) Una vez que dicho programa está en memoria, teclee las líneas de las distintas subrutinas propias del ordenador específico tal y como vienen impresas en este capítulo.
3) Salve todo en cinta o disquete.
4) (Facultativo) Considere que el programa es demasiado sencillo o poco eficaz y vuelva a escribir diez versiones, a cual mejor.

Descripción de las distintas subrutinas

LINEAS 10-30

Esta es la primera subrutina musical. Suena una nota de valor N y una duración L. Vamos a explicar mejor el significado de estos parámetros.

- N es un número entero de 0 a 96. Cada valor dista un semitono de los dos valores que lo rodean. La numeración elegida coincide con la del MSX-BASIC y la del GW-BASIC de los compatibles IBM. De esta manera, el valor 36 corresponde al DO central (o DO3 ya que pertenece a la tercera octava), 37 al DO#3, 38 a RE3, etc. (véase tabla 4.1). Una diferencia de 12 corresponde al intervalo de una octava. El área musical que abarca es de ocho octavas y cubre el registro de todos los ordenadores que examinamos. El valor 0 está reservado para la producción de una pausa de duración L.

Octava / NOTA	0	1	2	3	4	5	6	7
Do		12	24	36	48	60	72	84
Do #	1	13	25	37	49	61	73	85
Re	2	14	26	38	50	62	74	86
Re #	3	15	27	39	51	63	75	87
Mi	4	16	28	40	52	64	76	88
Fa	5	17	29	41	53	65	77	89
Fa #	6	18	30	42	54	66	78	90
Sol	7	19	31	43	55	67	79	91
Sol #	8	20	32	44	56	68	80	92
La	9	21	33	45	57	69	81	93
La #	10	22	34	46	58	70	82	94
Si	11	23	35	47	59	71	83	95

Al comienzo de la subrutina un test controla que la nota solicitada no supere las posibilidades del ordenador. Si esto ocurriera la nota se sustituiría por una pausa de igual duración.

- L es la duración de la emisión sonora, medida en dieciseisavos de segundo con alguna variación de ordenador a ordenador; lo importante es la relación de las duraciones, que es exacta en todas las implementaciones.

Podemos explicar esto mismo de otra manera: si escogemos un tiempo que corresponde a 120 negras por minuto, L es la duración del sonido expresada en número de fusas: L = 2 corresponde a una semicorchea, L = 4 a una corchea, L = 8 a una negra, etc.

LINEA 40

"Limpia" el vídeo (en modo texto) y sitúa el cursor arriba a la izquierda.

LINEA 50

Sitúa el cursor en la posición X de la línea en curso.

LINEAS 60-70

Esta es la otra subrutina musical. La subrutina de las líneas 10-30 traducía una concepción clásica de la música (sonaban notas de tonos bien definidos); ésta, en cambio, se presta a interpretaciones más libres: una vez llamada esta subrutina suena una nota de altura y duración totalmente definibles.

La altura debe ser expresada en Hz y puede ser un número cualquiera dentro de la gama de las frecuencias audibles y de los límites específicos de cada ordenador. La posibilidad de elección es más amplia que en la subrutina de las líneas 10-30 y no se limita a los valores presentes en una determinada escala musical. Para obtener una pausa basta con escoger una frecuencia nula.

La duración se expresa en ochentavos de segundo. La unidad es, por tanto, aproximadamente cinco veces menor que la utilizada en la primera subrutina y permite una mejor definición a la hora de elegir la longitud de la emisión sonora. Esta segunda subrutina musical será utilizada tan sólo en el capítulo séptimo y puede ser omitida cuando sean tecleados los otros programas.

LINEA 10000

Inicialización. Prepara la pantalla (en modo texto), llama a la subrutina de la línea 40.

Según el modelo, el ordenador puede ejecutar alguna que otra función: inicialización de los parámetros del sonido, lectura de la tabla de las notas, implementación de un pequeño programa en lenguaje máquina, etc.

Las subrutinas

Hemos tratado de construir estas subrutinas para los ordenadores más difundidos en el mercado, aunque ha sido trabajoso debido al caos existente entre los distintos BASICs y sus instrucciones musicales. Aquellas personas cuyos modelos no figuren en este apartado, si conocen el BASIC específico de su ordenador, deberían ser capaces de construir las subrutinas que deseen, pues dispondrán de muchos ejemplos en que basarse.

Aunque hemos procurado tener el máximo cuidado en todos los casos, hay que aceptar la posibilidad de que se cuele algún gazapo; les rogamos, si esto se produce, que nos disculpen.

SPECTRUM

```
10 IF N<1 OR N>96 THEN LET N=100
20 BEEP L*.062,N-36
30 RETURN
40 CLS:RETURN
```



```
50 PRINT TAB X;" ";:RETURN
60 LET F1=17.31234*LOG(F)-96.3763
65 BEEP L*.012,F1
70 RETURN
10000 GOSUB 40
10010 RETURN
```

¡Atención! Todas las instrucciones de asignación (tipo "N = 36") deben llevar la palabra LET (como LET "N = 36") según la tradición vigente en la firma Sinclair.

Otra particularidad: la palabra RND (1) del BASIC "estándar" se transforma en RND en el BASIC Spectrum.

APPLE II +, e, c

```
10 IF N<4 OR N>54 THEN N=4
15 IF L>25 THEN L=25
20 POKE 766,L*10:POKE 767,N-4:CALL 38144
30 RETURN
40 HOME:RETURN
50 HTAB X:RETURN
60 F1=10725/F:IF F1>255 THEN F1=0
65 POKE 6,F1:POKE 7,F1:POKE 9,255-2*L:CALL 38147
70 RETURN
10000 HIMEM:38144
10010 TEXT:HOME
10100 FOR I=0 TO 228
10110 READ A:POKE 38144 +I,A
10120 NEXT
10130 POKE 765,32
10200 RETURN
11000 DATA 76,55,149,164,7,173,48,192,230,8,208,5,230,9
,208,5
11010 DATA 96,234,76,21,149,136,240,5,76,27,149,208,235
,164,6,173
11020 DATA 48,192,230,8,208,5,230,9,208,5,96,234,76,47,
149,136
11030 DATA 240,209,76,53,149,208,235,173,255,2,10,168,1
85,127,149,133
11040 DATA 6,173,253,2,74,240,4,70,6,208,249,185,127,14
9,56,229
11050 DATA 6,133,7,200,185,127,149,101,6,133,6,169,0,56
,237,254
11060 DATA 2,133,9,169,0,133,8,165,7,208,152,234,234,76
,112,149
11070 DATA 230,8,208,5,230,9,208,5,96,234,76,125,149,20
8,236,0
11080 DATA 0,246,246,232,232,219,219,207,207,195,195,18
4,184,174,174,164
11090 DATA 164,155,155,146,146,138,138,130,130,123,123,
116,116,109,110,103
11100 DATA 104,97,98,92,92,87,87,82,82,77,78,73,73,69,6
9,65
11110 DATA 65,61,62,58,58,54,55,51,52,48,49,46,46,43,44
,41
```

```
11120 DATA 41,38,39,36,37,34,35,3233,30,31,29,29,27,28,
26
11130 DATA 26,24,25,23,23,21,22,20,21,19,20,18,18,17,17
,16
11140 DATA 16,15,16,14,15
```

El Apple no dispone de la instrucción INKEY$. Las líneas del tipo

```
1000 R$=INKEY$
```

deben ser sustituidas por

```
1000 GET R$
```

(Naturalmente, la numeración de la línea no tiene por qué ser 1000).

IBM PC Y COMPATIBLES

```
10 IF N>84 THEN N=0
15 LZ=32/L
20 PLAY "N=N;L=LZ;"
30 RETURN
40 CLS:RETURN
50 PRINT TAB(X)" ";:RETURN
60 IF F<37 OR F>32767 THEN  F=32767
65 SOUND F,L/5
70 RETURN
10000 GOSUB 40:KEY OFF
10010 PLAY "T120MFML"
10020 RETURN
```

OLIVETTI M10

```
10 IF N<15 OR N>78 THEN N=0
20 SOUND F(N),3*L-1
30 RETURN
40 CLS:RETURN
50 PRINT TAB(X);" ";:RETURN
60 IF F1=122892 /F:IF F1<415 OR F1>15800 THEN F1=0
65 SOUND F1,.6*L-.2
70 RETURN
10000 GOSUB 40
10010 DIM F(78)
10100 FOR I=15 TO 78
10110 READ F(I)
10120 NEXT I
10130 F(0)=0
10200 SOUND ON
10300 RETURN
```


MSX

```
10 IF N>95 THEN N=0
15 LZ=32/L
20 PLAY "N=N;L=LZ;"
30 RETURN
40 CLS:RETURN
50 PRINT TAB(X);" ";:RETURN
60 F1=125000!/F:FA=INT(F1/256):FB=F1-256*FA:IF FA>255 T
HEN 68
62 SOUND 0,FB:SOUND 1,FA
64 SOUND 7,62
66 SOUND 8,15
68 FOR T=1 TO 10*L:NEXT T
70 SOUND 8,0
75 RETURN
10000 SCREEN 0:GOSUB 40
10010 FOR I=0 TO 13:SOUND I,0:NEXT I
10200 RETURN
```

ATARI 400, 800, 800XL Y 130XE

```
10 IF N<24 OR N>60 THEN N=0
15 IF N=0 THEN 25
20 SOUND 0,F(I),10,15
25 FOR Z=1 TO 50:NEXT Z
30 SOUND 0,0,0,0
35 RETURN
40 PRINT ")":RETURN
50 POSITION X,23:RETURN
60 F1=63360 /F:IF F1>255 THEN 64
62 SOUND 0,F1,10,15
64 FOR T=1 TO 10*L:NEXT T
66 SOUND 0,0,0,0
70 RETURN
10000 GRAPHICS 0
10010 DIN F(60)
10020 FOR I=0 TO 3
10030 SOUND 1,0,0,0
10040 NEXT I
10100 FOR I=24 TO 60
10110 READ F(I)
10120 NEXT I
10200 RETURN
11000 DATA 15800,14912,14064,13284,12538,11836,11172,10
544,9952
11010 DATA 9394,8866,8368,7900,7456,7032,6642,6269,5918
,5586,5272,4968
11020 DATA 4697,4433,4184,3950,3728,3516,3321,3134,2954
,2793,2636,2484
11030 DATA 2348,2216,2092,1975,1864,1758,1660,1567,1479
,1396,1318,1244
11040 DATA 1171,1103,1043,987,932,879,830,783,739,693,6
59,622
11050 DATA 587,554,523,493,466,439,415
```

COMMODORE 64

```
10 IF N>95 THEN N=0
20 POKE S,FB(N):POKE S+1,FA(N)
22 POKE S+4,33
25 FOR T=1 TO 50*L:NEXT T
27 POKE S+4,32
30 RETURN
40 PRINT CHR$(147):PRINT CHR$(19):RETURN
50 PRINT TAB(X)" ";:RETURN
60 F1=F*.06097:FA=INT(F1/256):FB=F1-256*FA
62 IF FA>255 THEN 68
64 POKE S,FB:POKE S+1,FA
66 POKE S+4,33
68 FOR T=1 TO 10*L:NEXT T
70 POKE S+4,32
75 RETURN
10000 GOSUB 40
10010 S=54272 :POKE 649,1
10020 DIM FB(95),FA(95)
10100 FOR I=0 TO 95
10110 READ FB(I),FA(I)
10120 NEXT I
10200 FOR I=0 TO 26:POKE S+I,0:NEXT I
10210 POKE S+5,136:POKE S+6,248
10220 POKE S+24,15
10300 RETURN
11000 DATA 0,0,28,1,45,1,62,1,81,1,102,1,123,1,145,1,16
9,1,195,1,221,1,250,1,24,2
11010 DATA 56,2,90,2,125,2,163,2,204,2,246,2,35,3,83,3,
134,3,187,3,244,3,48,4
11020 DATA 112,4,180,4,251,4,71,5,152,5,237,5,71,6,167,
6,12,7,119,7,233,7,97,8
11030 DATA 225,8,104,9,247,9,143,10,48,11,218,11,143,12
,78,13,24,14,239,14,210,15,195,16
11040 DATA 195,17,209,18,239,19,31,21,96,22,181,23,30,2
5,156,26,49,28,223,29,165,31,135,33
11050 DATA 134,35,162,37,223,39,62,42,193,44,107,47,60,
50,57,53,99,56,190,59,75,63,15,67
11060 DATA 12,71,69,75,191,79,125,84,131,89,214,94,121,
100,115,106,199,112,124,119,151,126,30,134
11070 DATA 24,142,139,150,126,159,250,168,6,179,172,189
,243,200,230,212,143,225,,248,238,46,253
```

¡Cuidado! La instrucción R$=INKEY$ tampoco funciona en el Commodore y debe sustituirse por la instrucción GET R$

TEXAS TI99

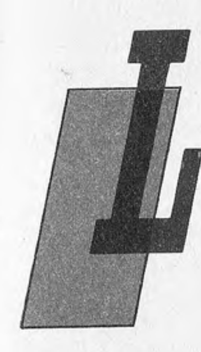
```
10 IF N<21 OR N>69 THEN N=0
20 CALL SOUND(60*L,F(N),30)
30 RETURN
40 CALL CLEAR:RETURN
50 PRINT TAB(X);" ";:RETURN
60 IF F<110 OR F>1760 THEN F=0
65 CALL SOUND(12*L,F,30)
```



```
70 RETURN
10000 GOSUB 40
10010 DIM F(69)
10100 FOR I=21 TO 69
10110 READ F(I)
10120 NEXT I
10130 F(0)=0
10200 RETURN
11000 DATA 110,117,123,131,139,147,156,165,175,185,196,
208,220
11010 DATA 233,247,262,277,294,311,330,349,370,392,415,
440
11020 DATA 466,494,523,554,587,622,659,698,740,784,831,
880
11030 DATA 932,988,1047,1109,1175,1245,1319,1397,1480,1
568,1661,1760
```

CAPITULO V

¡Y POR FIN, MUSICA!



legó el momento de pasar a la práctica... Nuestro objetivo, naturalmente, es el de interpretar música, pero hay muchas maneras de hacerlo con un ordenador; en este capítulo veremos dos de las más obvias. El primer programa nos permitirá tocar una pieza utilizando el teclado del ordenador; el segundo nos ofrece la misma posibilidad, pero esta vez la pieza musical habrá sido anteriormente almacenada en memoria.

Un miniórgano

Debemos confesar en primer lugar que este programa no representa un brillante comienzo: la realización del objetivo prefijado encuentra en seguida grandes dificultades debido a la escasa estandarización entre los distintos ordenadores.

En teoría, el hecho de escribir un programa que transforme un ordenador en órgano monofónico no resulta muy difícil: el corazón del programa es una subrutina que espera a que sea pulsada una tecla. Cuando esto ocurre el código de la tecla se traduce a una nota utilizando una tabla memorizada en la fase de inicialización y se emite el sonido correspondiente.

Los problemas surgen al tratar de traducir este sencillo procedimiento al BASIC estándar.

En primer lugar, para obtener un comportamiento parecido al de un verdadero teclado, la duración del sonido debería coincidir con la de presión de la tecla. Pero no existe una instrucción BASIC que permita controlar que una determinada tecla perma-

nezca pulsada. Esto es posible con muchos ordenadores (¡pero no con todos!) si hacemos PEEK en una determinada posición de memoria o utilizamos un pequeño programa en lenguaje máquina, pero es necesario escribir un programa específico para cada modelo y esto nos aleja de nuestros objetivos.

Para superar estas dificultades podemos optar por hacer sonar la nota elegida hasta que sea pulsada otra distinta (utilizando una tecla en concreto, por ejemplo SPACE, para apagar totalmente el sonido), pero nos encontramos frente a otro problema. En la gran mayoría de los BASICs presentes en los micros domésticos la instrucción que se encarga de la emisión de un sonido requiere como parámetro la duración del mismo: no es entonces posible tocar una nota hasta que se presente una nueva situación (en este caso la presión de una tecla). La solución consiste también en este caso en controlar el generador sonoro directamente en lenguaje máquina, pero para nuestro programa tuvimos que buscar algo diferente.

De esta manera se ha complicado algo la cosa. Cuando se pulsa una tecla la nota correspondiente se emite durante un tiempo muy breve y el control retorna al programa BASIC (en la línea 2010) para ver si ha sido apretada otra tecla. En caso negativo la misma nota seguirá sonando hasta que se registre la presión de una nueva tecla. De esta forma nos dará la impresión de que ha sido pulsada una sola nota durante más tiempo larga... o casi. En efecto, el tiempo que emplea el BASIC para los distintos controles comporta una brevísima interrupción del sonido. El resultado es un efecto de trémulo... que no resulta demasiado desagradable (¡e incluso podría parecer que lo hemos hecho a propósito!). La frecuencia del trémulo puede ser cambiada modificando el valor de la instrucción L = XX en la línea 2000.

Hay ya en el mercado programas de este tipo para la mayoría de los ordenadores personales y son, por regla general, mucho más eficaces que éste que publicamos aquí precisamente porque han sido concebidos para un ordenador en concreto. A pesar de ello lo incluimos para aquellos lectores que no tienen otro a su disposición y, sobre todo, porque su estructura es muy fácil de entender y se presta a todas las modificaciones que se quieran efectuar.

Modificaciones

En primer lugar, para un ordenador en particular pueden corregirse los puntos débiles que hemos mencionado anteriormente: gestión del teclado y control de la duración del sonido emitido.

Resulta fácil también aumentar la extensión sonora utilizando otras teclas además de las alfabéticas: bastará con modificar las líneas DATA en 20020-20140. Si queremos que la tecla con el código ASCII 60 corresponda a la nota con el código 56 (véase tabla del capítulo anterior) será necesario introducir el valor 56 en la posición 61 de la tabla DATA (en efecto, el primer valor corresponde al código 0).

Quienes dispongan de un generador de sonido no elemental podrán insertar una subrutina que cambie los parámetros del sonido producido (forma de onda, envolvente). Esta subrutina podrá ser llamada durante las fases de inicialización (antes de salir del subprograma en la línea 10000) o también durante la ejecución, utilizando teclas especiales (teclas de función o CTRL-XX).

Aquellos lectores que dispongan de un generador con varias voces querrán con toda seguridad transformar su ordenador en un órgano polifónico. Esta modificación no resulta tan sencilla, ya que hay que ser capaz de conocer los códigos de todas las teclas pulsadas en un mismo instante. Esto no siempre es posible y, en caso de que lo fuera, se requiere conocer bien la estructura interna del ordenador para escribir la subrutina en lenguaje máquina necesario.

Naturalmente resultará más fácil mejorar la presentación del programa añadiendo alguna subrutina gráfica que visualice la nota mientras suena...

Uso del programa

El manejo del programa resulta del todo claro: tras la inicialización el "instrumento" está listo para ser utilizado.

Las teclas utilizadas son las alfabéticas debido a un simple motivo: son las únicas que mantienen la misma posición en los teclados de todos los modelos de microordenadores (si son del tipo QWERTY naturalmente,). Como dijimos anteriormente, resulta fácil utilizar teclas adicionales.

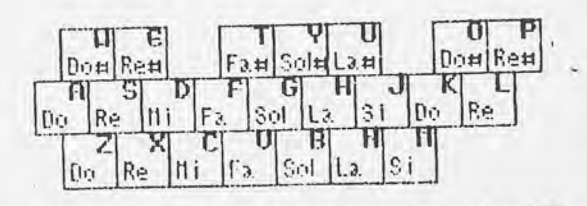


Figura 1.—Relación entre las teclas pulsadas y el sonido emitido por el ordenador.

La disposición adoptada aparece en la figura 1 y abarca algo más de dos octavas: la segunda octava está completa, mientras que la octava más baja no dispone de las notas alteradas.

Descripción del programa

¡No olvide incluir las subrutinas correspondientes del capítulo anterior!

1000-1100: Inicialización y título (muy sobrio...)
2000: Inicialización de las variables, nota (N) y longitud de la nota (L)
2010-2050: Ciclo principal. Si una tecla ha sido pulsada su valor se traduce en un código de nota (utilizando el vector NT(X)) y esta suena. En caso contrario se repite la nota anterior.
20000-20140: Lectura de la tabla de conversión código ASCII-código nota.

Listado 5.1

En los ordenadores que no disponen de la instrucción INKEY$ (C64, por ejemplo) el comando A$=INKEY$ de la línea 2010 debe ser sustituido por el comando GET R$.

En el caso concreto del Apple es necesario añadir o modificar las siguientes líneas como se indica:

```
1001 KB=49152 :KS=49168 :POKE KS,0
2010 A=PEEK(KB):IF A<128 THEN GOSUB 10:GOTO 2010
2020 POKE KS,0:A=A-128
```

En cualquier caso, ¡no hay que olvidar añadir las subrutinas del capítulo 4!

```
1 GOTO 1000
1000 GOSUB 10000
1010 PRINT:PRINT:PRINT:PRINT:PRINT
1020 PRINT"                    MINI-ORGANO"
1100 GOSUB 20000
2000 L=2:N=0
2010 A$=INKEY$:IF A$="" THEN GOSUB 10:GOTO 2010
2020 A=ASC(A$)
2050 N=NT(A):GOTO 2010
3000 END
20000 DIM NT(127)
20010 FOR I=0 TO 127:READ NT(I):NEXT I
```



```
20015 RETURN
20020 DATA 0,0,0,0,0,0,0,0,0,0
20030 DATA 0,0,0,0,0,0,0,0,0,0
20040 DATA 0,0,0,0,0,0,0,0,0,0
20050 DATA 0,0,0,0,0,0,0,0,0,0
20060 DATA 0,0,0,0,0,0,0,0,0,0
20070 DATA 0,0,0,0,0,0,0,0,0,0
20080 DATA 0,0,0,0,0,36,31,28,40,39
20090 DATA 41,43,45,0,47,48,50,35,33,49
20100 DATA 51,0,0,38,42,46,29,37,26,44
20110 DATA 24,0,0,0,0,0,0,36,31,28
20120 DATA 40,39,41,43,45,0,47,48,50,35
20130 DATA 33,49,51,0,0,38,42,46,29,37
20140 DATA 26,44,24,0,0,0,0,0
```

Minieditor musical

Este segundo programa tampoco se distingue precisamente por su originalidad, pero al menos no ofrece tantos problemas por la diferencia entre los ordenadores y sus lenguajes respectivos como el anterior. Se trata de introducir una pieza musical en memoria transcribiendo las indicaciones de una partitura o, simplemente, siguiendo lo que indica la inspiración del momento y hacer que el ordenador la interprete.

Tal y como está el programa funciona de manera satisfactoria, presentando incluso alguna sofisticación que facilita su uso, aunque también se podrán añadir unos últimos retoques y, como veremos más adelante, módulos para aumentar su potencia.

Los módulos más importantes son dos: el que permite introducir en memoria la pieza elegida y el que permite interpretarla. Otros módulos permiten listar o modificar las notas en memoria e incluso salvar o cargar una pieza utilizando cintas o disquetes.

La pieza se memoriza con la ayuda de dos vectores unidimensionales. El primero, N(X), contiene el código de las notas, y el segundo, L(X), su longitud o duración. La dimensión de estos vectores (y por tanto la longitud de la pieza musical) ha sido prefijada arbitrariamente en 1000. Si su ordenador tiene problemas de memoria este valor puede ser reducido. Cuando, en cambio, lo que desee sea memorizar toda una sinfonía, podrá aumentar ese número dentro de los límites de la memoria disponible; bastará para ello con modificar adecuadamente las líneas 1100 y 3100.

Modificaciones

Acabamos de hablar de la primera modificación: la dimensión de la tabla de las notas en memoria.

Es posible aportar alguna mejora a las posibilidades de edición en la fase de introducción de las notas. Podemos utilizar las teclas de desplazamiento del cursor para modificar un dato sin tener que reescribir toda la línea. Este tipo de posibilidad depende, naturalmente, del tipo de ordenador que se esté utilizando y es por eso que no lo hemos incluido en el programa.

Podemos también mejorar de alguna manera la seguridad del programa. Ya han sido incluidos numerosos controles sobre la entrada por parte del usuario, pero antes o después siempre puede ocurrir algo extraño... que produzca resultados inesperados. Es posible añadir una subrutina global del tipo ONERR GOSUB, pero debe ser diferente para cada ordenador.

Naturalmente, no habrá problema alguno a la hora de añadir cualquier modificación documental (títulos, presentación del menú y de los listados...).

Aquellas personas que posean un generador sonoro algo más evolucionado querrán añadir un módulo para modificar los parámetros del sonido (forma de onda, envolvente, etc.) antes de ejecutar la pieza.

Es lógico también que quienes tengan la posibilidad traten de adaptar el programa capacitándolo para la ejecución de piezas polifónicas. Para esto necesitamos dos o más vectores para las notas (N1(X), N2(X), etc.) y sus longitudes (L1(X), L2(X), etc.). Las modificaciones para la ejecución son bastante obvias. Para la introducción de las notas pueden seguirse dos vías. En la primera basta con repetir una subrutina análoga a la de las líneas 3000-3920 por cada voz. La ventaja es que las modificaciones que hay que aportar son mínimas, pero como contrapartida resulta fácil perder el sincronismo entre las diistintas voces, pudiendo obtener resultados desastrosos.

La otra solución requiere una mayor modificación de la subrutina para poder suministrar juntos los datos correspondientes a las distintas voces. El problema es que las notas pueden tener distinta duración según las voces. Entonces habrá que escoger la duración mínima como duración común y repetir en el índice las notas más largas. Surge entonces un problema al ejecutarlo: una nota larga es interpretada como una sucesión de varias notas de la misma altura. Es posible remediar esto, aunque no resulta muy sencillo; por ejemplo, añadiendo un valor suficiente (pongamos 100) a la duración de una nota que es introducida en varios trozos y utilizar esta indicación en la fase de ejecución.

Uso del programa

Después del título aparece el menú principal. Vamos a describir las siguientes opciones:

1) CREAR MUSICA

Tendremos que introducir línea a línea las sucesivas notas de la pieza. Para cada nota hay que precisar (después del número de orden, impreso automáticamente) su nombre (DO, RE, MI...), la octava (teniendo en cuenta los límites del ordenador) y la longitud. Para pasar de un campo a otro será necesario pulsar la barra espaciadora.

Para el nombre de la nota será suficiente con escribir la primera letra, pues el programa proporciona por sí solo la segunda. Esto, naturalmente, no vale para SI y SOL: después de una "S" el programa espera la introducción de una "O" o de una "I". Tras el nombre de la nota se podrá teclear + (para incluir un sostenido) o - (para un bemol). Para repetir la nota precedente bastará con apretar SPACE y la nota será incluida automáticamente en memoria y en la pantalla. Para poner una pausa será necesario teclear "P" (en este caso la octava no tiene importancia alguna).

Después se pulsa SPACE y se escribe un número del 1 al 8 (o menos, según las posibilidades del ordenador) que representa la octava. También en este caso si la octava no ha cambiado en relación con la nota precedente (que será el caso más frecuente) bastará con apretar SPACE.

El último dato es la duración de la nota; se indica por un número de la siguiente manera:

1 — Fusa
2 — Semicorchea
3 — Corchea
4 — Negra
5 — Blanca
6 — Redonda
etc.

Si estas palabras le traen pocos recuerdos a la memoria basta saber que cada número corresponde a una nota cuya duración es doble que la de la anterior.

Ahora hay dos posibilidades: pulsar RETURN para pasar a la línea y nota siguiente o pulsar de nuevo "R" para incluir otra vez los datos de la misma nota en caso de error.

Para terminar la introducción de las notas y volver al menú principal habrá que apretar la tecla ESCAPE desde el comienzo de la línea.

2) LISTADO DE LA PIEZA

Será suficiente con meter el número de la primera nota y el de la última para que la pieza en memoria sea decodificada en un formato análogo al utilizado en la fase de introducción de los datos.

Si se aprieta RETURN en respuesta a la segunda pregunta, el listado seguirá adelante hasta la última nota de la pieza.

3) MODIFICACION DE LA COMPOSICION

Hay que especificar el número de la primera nota que hay que modificar (se obtiene listando la pieza). Los datos de la nota se incluyen igual que en la primera opción.

Las modificaciones siguen para las notas sucesivas hasta que se pulse ESCAPE.

Esta opción permite también añadir notas a una pieza que se halla en memoria.

4) INTERPRETAR MUSICA

Se ejecuta la pieza en memoria.

5) y 6) SALVAR Y CARGAR UNA PIEZA

Estos módulos no están codificados en el listado, debido a un simple motivo: no sólo el código es diferente para cada ordenador, sino que depende también del tipo de memoria de masa utilizada (cintas, disquetes, micro-drives...).

El objetivo de estos módulos es la grabación permanente y la recuperación de una pieza introducida anteriormente en memoria.

No resulta difícil instalarlos para un ordenador en concreto. El módulo 5 sólo debe salvar los dos vectores N(X) y L(X). Antes de esta operación la longitud de la pieza se pone en N(0) y L(0) con instrucciones N(0) = MAX y L(0) = MAX. Esto permite salvar sólo la parte útil de la pieza y no los 1000 elementos del vector siempre.

El módulo 6 es parecido, pero carga desde la memoria de masa los vectores N(I) y L(I), tras haber puesto N(0) en MAX.

Descripción del programa

1000-1220: Inicialización y título (siempre sobrio...)
2000-2250: Menú principal y saltos a los distintos módulos.
3000-3920: Introducción de la pieza. Desde la línea 3100 sirve también para la subrutina de modificación de la pieza. El programa reconoce la subrutina activa por la posición de la bandera F (flag en inglés). F = 0 para la opción número 1, F = 1 para la opción número 3.

3100-3130: Incrementa el índice y vuelve al menú si se supera el número máximo de las notas disponibles.
3200-3410: Introducción y visualización de la nota (nótese el caso particular R$ = "S").
3420-3480: Inserción, si se da el caso, de sostenidos o bemoles.
3500-3580: Introducción de la octava y cálculo del código de la nota (en la línea 3530).

El test N < >-1 permite dar el código 0 cuando ha sido incluida una pausa. En efecto, N < >-1 vale 0 cuando N vale -1. Puede parecer extraño que se aplique la función ABS a N< >-1. No todos los ordenadores dan el mismo valor a una variable booleana cuando el test es positivo. Este valor puede ser 1 ó -1, de aquí nace la necesidad de usar el valor absoluto.

3600-3680: Introducción y codificación de la duración.
3700-3800: Fin de la introducción de datos. La ejecución del programa vuelve a 3100 para una nueva nota o a 3200 para meter de nuevo la misma.
3900-3920: Salida del módulo.
4000-4210: Ejecución de la pieza.
5000-5610: Módulo de listado de la pieza.
5100-5130: Control de la presencia de una pieza en memoria.
5200-5240: Introducción de las líneas iniciales y finales y control de la validez de las respuestas.
5300-5520: Decodificación de los valores en las tablas de notas y visualización del listado.
6000-6130: Módulo de modificación de la pieza en memoria. Después de la introducción (y el control) de la línea inicial el flujo del programa está orientado hacia la línea 3100.
7000 y 8000: Módulos de salvaguardia y carga de los datos.

LISTADO 5.2

El programa utiliza a menudo la instrucción INKEY$. Véanse las indicaciones dadas en el capítulo 4 para las sustituciones que hay que efectuar en ordenadores que no dispongan de dicha instrucción.

```
1 GOTO 1000
1000 GOSUB 10000
1100 DIM L(1000),N(1000)
1110 MAX=0:F=0
1200 PRINT:PRINT:PRINT:PRINT:PRINT:PRINT
1210 PRINT "                MUSICA..."
```

```
1220 R$=INKEY$:IF R="" THEN 1220
2000 GOSUB 40
2100 PRINT:PRINT:PRINT:PRINT
2110 PRINT " 1. CARGAR MUSICA"
2120 PRINT " 2. LISTADO DE LA PIEZA"
2130 PRINT " 3. MODIFICACION DE LA PIEZA"
2140 PRINT " 4. INTERPRETAR MUSICA"
2150 PRINT " 5. SALVAR UNA PIEZA"
2160 PRINT " 6. CARGAR UNA PIEZA"
2170 PRINT " 7. FIN":PRINT
2200 INPUT R$
2210 R=VAL(R$)
2220 IF R<1 OR R>7 THEN 2000
2250 ON R GOTO 3000,5000,6000,4000,7000,8000,2500
2500 END
3000 GOSUB 40
3010 I=0:F=0
3100 PRINT:I=I+1:IF I<1001 THEN 3200
3110 PRINT " MEMORIA AGOTADA "
3120 R$=INKEY$:IF R$="" THEN 3120
3130 GOTO 2000
3200 X=1:GOSUB 50
3210 PRINT I;:X=5:GOSUB 50:PRINT "NOTA:";
3220 R$=INKEY$:IF R$="" THEN 3220
3230 R=ASC(R$)
3240 IF R=27 THEN 3900
3250 IF R=32 THEN PRINT N$;:GOTO 3500
3300 IF R$="D" THEN N$="DO":N1=0:GOTO 3400
3310 IF R$="R" THEN N$="RE":N1=2:GOTO 3400
3320 IF R$="M" THEN N$="MI":N1=4:GOTO 3400
3330 IF R$="F" THEN N$="FA":N1=5:GOTO 3400
3340 IF R$="L" THEN N$="LA":N1=9:GOTO 3400
3345 IF R$="P" THEN N$="-":N1=-1:PRINT N$;:GOTO 3480
3350 IF R$<>"S" THEN 3220
3360 PRINT R$;
3370 R$=INKEY$:IF R$="" THEN 3370
3380 IF R$="O" THEN N$="SOL":N1$="OL":N1=7:GOTO 3400
3390 IF R$="I" THEN N$="SI":N1$="I":N1=11:GOTO 3400
3395 GOTO 3370
3400 PRINT N1$;
3410 R$=INKEY$:IF R$="" THEN 3410
3420 R=ASC(R$)
3430 IF R=32 THEN 3500
3440 IF R=43 THEN N1=N1+1:N$=N$+"+":GOTO 3470
3450 IF R<>45 THEN 3410
3460 N1=N1-1:N$=N$="-"
3470 PRINT R$;
3480 R$=INKEY$:IF R$="" THEN 3480
3500 X=15:GPSUB 50
3505 PRINT "OT:";
3510 R$=INKEY$:IF R$="" THEN 3510
3520 R=ASC(R$)
3530 IF R=32 THEN N=ABS(N1<>-1)*(O*12+N1):PRINT O$;:GOT
O 3600
3540 IF R<49 OR R>56 THEN 3510
3550 O=R-48:O$=R$
3560 N=ABS(N1<>-1)*(O*12+N1)
```



```
3570 PRINT O$
3580 R$=INKEY$:IF R$="" THEN 3580
3600 X=22:GOSUB 50
3610 PRINT "DUR:";
3620 R$=INKEY$:IF R$="" THEN 3620
3630 R=ASC(R$)
3640 IF R=32 THEN 3680
3650 IF R<49 OR R>56 THEN 3620
3660 L=R-48:L$=R$
3670 L=2+(L-1)
3680 PRINT L$;
3700 R$=INKEY$:IF R$="" THEN 3700
3710 R=ASC(R$)
3720 IF R=32 THEN 3200
3730 IF R<>13 THEN 3700
3740 N(I)=N:L(I)=L
3800 GOTO 3100
3900 IF F=0 THEN MAX=I-1:GOTO 2000
3910 IF I-1>MAX THEN MAX=I-1
3920 GOTO 2000
4000 GOSUB 40
4100 FOR I=1 TO MAX
4110 N=N(I):L=L(I)
4120 GOSUB 10
4130 NEXT
4210 GOTO 2000
5000 GOSUB 40
5040 GOTO 2000
5100 IF MAX<>0 THEN 5200
5110 PRINT "     NO HAY NINGUNA"
5115 PRINT "     PIEZA EN MEMORIA"
5120 R$=INKEY$:IF R$="" THEN 5120
5130 GOTO 2000
5200 PRINT " LINEA INICIAL ";
5210 INPUT R$:R1=VAL(R$)
5212 IF R1<1 OR R1>MAX THEN GOSUB 40:GOTO 5200
5220 PRINT " LINEA FINAL ";
5230 INPUT R$:R2=VAL(R$)
5232 IF R$="" THEN R2=MAX
5234 IF R2<1 OR R2>MAX THEN GOSUB 40:GOTO 5220
5240 PRINT
5300 FOR I=R1 TO R2
5310 PRINT I;
5320 N=N(I):IF N=0 THEN N$="-":GOTO 5450
5325 O=INT(N/12):N1=N-12*O
5330 IF N1=0 THEN N$="DO"
5340 IF N1=1 THEN N$="DO+"
5350 IF N1=2 THEN N$="RE"
5360 IF N1=3 THEN N$="RE+"
5370 IF N1=4 THEN N$="MI"
5380 IF N1=5 THEN N$="FA"
5390 IF N1=6 THEN N$="FA+"
5400 IF N1=7 THEN N$="SOL"
5410 IF N1=8 THEN N$="SOL+"
5420 IF N1=9 THEN N$="LA"
5430 IF N1=10 THEN N$="LA+"
5440 IF N1=11 THEN N$="SI"
```

```
5450 X=5:GOSUB 50:PRINT "NOTA:";:PRINT N$;
5460 X=15:GOSUB 50
5470 PRINT "OT:";:PRINT O;
5480 L=L(I)
5490 L=INT(10*LOG(L)/LOG(2)+.1))/10+1
5500 X=22:GOSUB 50
5510 PRINT "DUR:";:PRINT L
5520 NEXT I
5600 R$=INKEY$:IF R$="" THEN 5600
5610 GOTO 2000
6000 GOSUB 40
6010 IF MAX<>0 THEN 6100
6020 PRINT "        NO HAY NINGUNA"
6030 PRINT "        PIEZA EN MEMORIA"
6100 PRINT " LINEA INICIAL ";
6110 INPUT R$:R=VAL(R$)
6120 IF R<1 OR R>MAX THEN GOSUB 40:GOTO 6100
6130 I=R-1:F=1:GOTO 3100
7000 GOTO 2000
8000 GOTO 2000
```


CAPITULO VI

EL ORDENADOR COMO COMPOSITOR (CONVENCIONAL)

Gracias a los programas del capítulo anterior nuestro ordenador es capaz de emitir música o, mejor dicho, nosotros conseguimos que toque, ya que hasta ahora se ha limitado a ejecutar fielmente las melodías que le hemos indicado, sin dar pruebas de tener iniciativa alguna.

Pero más de una vez hemos oído hablar de música compuesta por ordenadores... ¿Es posible lograr eso mismo con nuestros medios tan modestos? ¡Naturalmente! Trataremos ahora de escribir algún programa capaz de producir piezas musicales de tipo convencional (clásico o ligero), mientras que en el próximo capítulo nos adentraremos ya más en el campo de la experimentación.

Primero un comentario forzoso: casi desde que existen los ordenadores ha habido intentos de utilizarlos para componer música, pero ¿sabe de alguna composición computerizada de estilo convencional (tal vez una sinfonía o una simple canción) que haya sido reconocida, no ya como obra maestra, sino como una creación, al menos, lograda? Lo dudamos mucho.

Podríamos responder, por un lado, que los músicos utilizan todo su peso para quitarle mérito a una composición de este tipo, pero hay otra explicación más sencilla: no es posible hacer una comparación objetiva entre una obra escrita por un ordenador y una composición humana. Ni siquiera los mismos estudiosos de la música computerizada son capaces de negar la superioridad del hombre sobre la máquina en un tema de este tipo.

Sería posible hallar distintos argumentos más o menos filosóficos para apoyar esto, pero nos limitaremos a hacer una simple consideración: aunque se ha comparado a menudo al ordenador

con un cerebro humano, este último presenta unas características muy peculiares. La estructura en sí del cerebro, con una interconexión de millones de células, presenta una complejidad bastante superior a la de los más potentes ordenadores de hoy en día. Hay, además, otra diferencia fundamental: el cerebro se desarrolla y evoluciona debido a una serie de experiencias exteriores que influyen sobre su funcionamiento. Una cosa es transformar un ordenador en un excelente jugador de ajedrez, disciplina con reglas precisas en las que la experiencia (partidas anteriores) es fácilmente numerable, y otra tratar de hacer de él un experto en composición musical. Sí, también en este caso hay reglas, pero su interpretación es mucho más libre; además, en el momento de la creación hay experiencias muy diversas (y no sólo de naturaleza musical) que pueden influir de forma determinante sobre el resultado final. El ordenador no posee una historia personal y esto explica la dificultad que existe a la hora de imitar al hombre en las creaciones artísticas y culturales.

Naturalmente, con los progresos de la tecnología la situación variará, pero esto no pasará antes de que transcurran bastantes años.

Además ¿por qué utilizar el ordenador para copiar sin mucho éxito una actividad en la que el hombre no se las arregla tan mal, cuando hay tantos campos nuevos en los que la participación de la máquina resulta indispensable? El hecho de programar un ordenador para componer música convencional difícilmente dará resultados muy satisfactorios. Una orientación mucho más provechosa consiste en experimentar nuevas vías y en utilizar las extraordinarias posibilidades que éste nos ofrece para crear una música diferente e inconcebible con los medios tradicionales.

Un último consejo: no trate de escribir la nueva "canción del verano" para esta temporada o de componer la décima sinfonía de Beethoven. Si se quita de la cabeza estas ilusiones podrá hallar cosas interesantes en los programas e ideas que aparecen en este capítulo, que podrán además ayudarle a entender los mecanismos de la composición musical.

Primeras composiciones

Para que el ordenador comience a componer bastará con introducir el cortísimo programa que incluimos a continuación (sin olvidar, naturalmente, añadir las subrutinas del capítulo 4). Este programa produce notas con valor y duración elegidas arbitrariamente por el ordenador (en esta ocasión nos hemos limitado a variaciones de frecuencia en el intervalo de dos octavas).


```
2000 N=24+INT(24*RND(1))
2100 L=2+INT(6*RND(1))
2200 GOSUB 10:GOTO 2000
```

Bueno, está claro que resulta bastante molesto escuchar notas que saltan continuamente de una frecuencia a otra. Probemos ahora con otro programa con resultados un poco menos irritantes: el tono de la nota se escoge siempre aleatoriamente, pero dentro de un intervalo reducido (4 semitonos como máximo) en torno a la nota precedente. El resultado puede escucharse algo mejor, pero sin duda alguna hay que atreverse con programas más depurados.

```
2000 N=36:L=4:GOSUB 10
2100 DN=INT(9*RND(1))-4:N1=N+DN
2110 IF N1>48 OR N1<24 THEN N1=N-DN
2120 N=N1
2200 L=2+2*INT(3*RND(1))
2300 GOSUB 10
2400 GOTO 2100
```

Veamos como último ejemplo de programa de composición casual el que permite que el ordenador interprete... una parte de su memoria. La duración y tono de las notas se obtiene de posiciones sucesivas de memoria, tras una necesaria reconversión. Se solicitan las direcciones de partida y final de la zona que va a musicalizarse. A menudo el resultado desilusiona, pero puede también resultar divertido: ¡trate de "tocar" una zona de variables, la ROM o, mejor, una imagen gráfica cargada en memoria!

Este programa tal vez acabe siendo menos inútil de lo que parece a primera vista: con un poco de costumbre puede utilizarlo para comprobar rápidamente la memoria del ordenador...

```
2000 GOSUB 40
2100 PRINT "DIRECCION DE PARTIDA:";
2110 INPUT R$
2120 A1=VAL(R$)
2200 PRINT "DIRECCION FINAL      :";
2210 INPUT R$
2220 A2=VAL(R$)
2300 D=A2-A1:IF D<0 THEN 2100
2310 D1=INT(D/2)
2320 FOR I=0 TO D1-1
2330 N=24+INT(PEEK(A1+I)/10)
2340 L=2+2*INT(PEEK(A1+D1+I)/50))
2350 GOSUB 10
2360 NEXT I
3000 END
```

Experimentemos seriamente

Ya sabíamos que un ordenador no hace nada por sí mismo y que los resultados dependerán del programa que hayamos introducido. Las experiencias anteriores han podido desilusionar, ya que toda composición musical obedece a ciertas reglas que difícilmente pueden seguir un conjunto de notas elegidas al azar.

Desde que se trabaja en el campo de la composición "automática" se han seguido dos vías principales para producir piezas que recuerden a las creaciones humanas.

La primera consiste en codificar con precisión las reglas de composición a las que obedece cierto tipo de música, sirviéndose del análisis de un gran número de obras. El programa crea la pieza siguiendo estas reglas y prevé un mínimo de elecciones casuales para no dar siempre los mismos resultados.

La segunda trata de considerar la sucesión de las notas como un proceso de Markov. No vamos a profundizar sobre este tema, pero trataremos de ilustrar al menos su aplicación en la creación musical. Cada elemento de una composición está estrechamente ligado a los que lo preceden: en las melodías escritas en DO mayor, tras un DO habrá con más frecuencia un MI que un FA. En otras palabras, la posibilidad de encontrar una nota determinada depende de la anterior. Al conocer la probabilidad de cada nota podremos construir paso a paso una pieza que respete estos datos.

En la práctica las cosas son algo más complicadas: el hecho de decidir una nota sólo a partir de la precedente resulta muy restrictivo: habrá que tener en cuenta las dos, tres o más notas anteriores. Se requiere entonces conocer las probabilidades ligadas a cada pareja, terceto, etc. de notas sucesivas, lo que complica no poco el asunto.

Por otra parte, hay otros parámetros además del tono de la nota que caracterizan una composición, empezando por la duración: ésta podrá también obedecer a una elección de probabilidad. Si queremos, además, que haya varias notas a un tiempo, la situación se hace especialmente complicada.

Para conocer las probabilidades ligadas a cada sucesión de notas el método más sencillo consiste en utilizar resultados estadísticos obtenidos al analizar piezas de música "verdadera". Naturalmente, cada tipo de música dará tablas de probabilidad distintas. Este método permite componer, según se prefiera, música al estilo de Mozart, de Schonberg o de Falla. Si queremos, ¡podemos hasta componer música 35% Mozart, 65% Falla!

Como puede verse, la ventaja de este método es la de componer música de cualquier tipo sin necesidad de un gran trabajo de análisis de dicha música. El lado negativo es que a una pieza

producida de esta manera le falta la estructura de conjunto que una composición humana posee.

Poniendo en práctica el método Markov

Vamos a poner en práctica el método que acabamos de describir (programa del listado 6.1).

La composición producida es monofónica y cada nota está determinada por dos parámetros: tono y duración.

El programa se ha escrito para integrarlo en el del listado 5.2 del capítulo anterior. El listado 6.1 contiene sólo las líneas que hay que añadir (o modificar) respecto al listado 5.2 para obtener el programa completo. La pieza sobre la que se apoya el análisis estadístico debe estar en memoria en forma de los ya conocidos vectores N(X) y L(X), que pueden ser asignados en el momento o cargados desde cinta o disquete. Bastará con fijar la longitud de la obra y, si se desea, las primeras notas y esperar el resultado. Naturalmente si el resultado está bien logrado podrá salvarlo para que lo escuchen sus descendientes.

Hemos determinado que se tengan en cuenta sólo las dos notas precedentes a la nota que hay que crear. Con tres notas o más el resultado puede ser mejor, pero si la pieza que hay que analizar no es lo suficientemente larga corremos el riesgo de hallar a menudo secuencias de tres (o más) notas que no estarán nunca presentes en la pieza. Aun así puede tratar de modificar el programa en este sentido.

El mecanismo de creación de la obra es sencillo: el programa toma en consideración los códigos que corresponden a la altura de las notas anteriores a la que hay que producir (línea 9415). Esta sucesión de dos códigos se busca en la tabla N(X) en memoria (líneas 9430-9440). Cuando ha sido identificada se incrementa en la tabla SN con el valor que corresponde al código de la nota que sigue a las dos notas buscadas (línea 9450). Al final de esta primera fase la tabla SN contiene los mayores valores de probabilidad para la aparición de las notas correspondientes. Después de esto la elección se efectúa de forma que se respeten estas probabilidades utilizando un número aleatorio generado por el ordenador (líneas 9470-9495). Si tal secuencia no existe en la melodía verdadera, el tono se elige arbitrariamente en las cercanías del precedente (líneas 9462-9468).

La altura calculada se pone en la tabla NN(X) y se repite el mismo proceso para la elección de la duración, que hay que memorizar en la tabla LL(X) (líneas 9520-9596).

Naturalmente, el procedimiento sería mucho más rápido si pudiéramos construir con anterioridad dos tablas (una para el tono

y otra para la duración) en las que estuvieran reseñadas las probabilidades de aparición de cada nota en dependencia de las dos notas precedentes. Por desgracia, estas tablas, incluso limitándose al examen de las dos notas anteriores y dentro de un intervalo de dos octavas, ocuparían no menos de 50.000 posiciones de memoria, cosa que resulta un poco excesiva para la mayoría de los micros domésticos.

Uso del programa

Para disponer del programa completo hay que proceder en dos tiempos: primero hay que cargar el programa del capítulo anterior (listado 5.2) y, a continuación, teclear las líneas que aparecen en el listado 6.1. El resultado será, naturalmente, salvado sobre soporte magnético (con el nombre de Markov, por ejemplo).

El menú principal contiene dos nuevas opciones, vamos a verlas:

8) COMPOSICION DE UNA PIEZA

Es el módulo de creación de una nueva pieza musical por parte del ordenador. Esto será posible sólo si se halla ya en memoria la obra de referencia. Podemos meterlo "a mano" utilizando la opción 1) o cargarlo desde cinta o disquete utilizando la opción 6).

La pantalla muestra la primera pregunta, que afecta a la longitud de la pieza que va a ser creada. No hay que olvidar que una pieza con una longitud de un centenar de notas puede requerir varios minutos para su creación.

Luego pregunta si se desean escoger las dos primeras notas de la composición. Si la respuesta es positiva habrá que modificar la altura y duración de las dos notas según el convenio establecido en el capítulo 4 (el ordenador precisa el intervalo de las elecciones compatibles con la pieza de referencia). En caso de respuesta negativa el ordenador escoge al azar estos datos con las mismas verificaciones de compatibilidad.

Después de esto el ordenador se pone a trabajar y basta con esperar los resultados.

Cuando el menú principal vuelve a aparecer en la pantalla, la nueva creación estará presente en memoria en los vectores NN (altura) y LL (duraciones). Para escucharla habrá que utilizar la opción 7).

7) INTERCAMBIO DE COMPOSICIONES EN MEMORIA

Se intercambian las parejas de vectores N-L y NN-LL. Las opciones de la 2 a la 6 se aplican siempre a la obra musical memorizada en N-L. Tras un primer intercambio estos vectores contienen la pieza creada por el ordenador: es posible hacerlo sonar (opción 4), listarlo (opción 2), modificarlo (opción 3) o salvarlo sobre un soporte magnético (opción 5). Si utilizamos por segunda vez la opción 7 la melodía de referencia ocupará de nuevo los vectores N-L y podrá ser utilizada para crear otras piezas musicales.

Modificaciones

La principal modificación que será posible efectuar sin alterar el programa es la del número de notas precedentes que hay que tomar en consideración. Bastará con algún cambio en las líneas 9200-9388 (para la elección de las notas iniciales) y en las líneas 9415-9460 y 9515-9560 (para la elección de las notas sucesivas).

Es posible también tratar de extender el proceso a otros parámetros (volumen sonoro...) sin grandes problemas. El hecho de escribir composiciones polifónicas es algo totalmente distinto y es mejor volver a escribir desde cero todo el programa. Resultan más sencillas las mejoras estéticas: presentación, opción para la elección de los parámetros del sonido, etc.

Descripción del programa

Para entender el funcionamiento del programa hay que volver a leer atentamente las explicaciones dadas más arriba.

1100-1110: Inicialización de vectores y variables..
2165-2250: Modificación del menú principal.
4500-4700: Módulo de intercambio de los vectores N-L y NN-LL.
9000-9030: Entrada de la longitud de la pieza.
9100-9160: Cálculo del intervalo de las alturas y de las duraciones representadas en la melodía de referencia.
9200-9290: Entrada de los parámetros de las dos notas iniciales, con sus comprobaciones respectivas.
9300-9386: Elección casual de las dos primeras notas por parte del ordenador, con sus comprobaciones respectivas.
9400-9405: Inicialización de los vectores NN y LL.
9410: Inicio del bucle, repetido para cada nota de la pieza que hay que crear.
9420-9460: Construcción de la tabla de probabilidades de aparición de una nota, dadas las dos notas precedentes..
9462-9468: Si la secuencia de dos notas no ha aparecido nun-

ca en la melodía de referencia, el ordenador escoge una nota arbitraria en las cercanías de las dos anteriores.

9470-9495: La nota se elige de forma casual, según las probabilidades de la tabla construida en 9420-9460.

9500-9595: El mismo procedimiento se repite a la hora de elegir la duración de la nota (véase comentarios de las líneas 9400-9495).

9596-9599: Finalización del bucle iniciado en la línea 9410 y retorno al menú principal.

LISTADO 6.1

```
1100 DIM L(1000),N(1000),NN(1000),LL(1000),EN(96),EL(96
)
1110 MAX=0:F=0:LP=0
2165 PRINT " 7. INTERCAMBIO DE PIEZAS EN MEMORIA "
2167 PRINT " 8. COMPOSICION DE UNA PIEZA"
2170 PRINT " 9. FIN":PRINT
2220 IF R<1 OR R>9 THEN 2000
2250 ON R GOTO 3000,5000,6000,4000,7000,8000,4500,9000,
2500
4500 FOR I=1 TO 1000
4510 T=N(I):N(I)=NN(I):NN(I)=T
4520 T=L(I):L(I)=LL(I):LL(I)=T
4530 NEXT I
4600 T=MAX:MAX=LP:LP=T
4700 GOTO 2000
9000 GOSUB 40
9010 PRINT " LONGITUD DE LA PIEZA  (3-1000) :  ";
9020 INPUT LP
9030 IF LP<3 OR LP>1000 THEN 9010
9100 M1=96:M2=0:M3=8:M4=I
9110 FOR I=1 TO MAX
9120 IF N(I)<M1 AND N(I)<>0 THEN M1=N(I)
9130 IF N(I)>M2 THEN M2=N(I)
9140 IF L(I)<M3 THEN M3=L(I)
9150 IF L(I)>M4 THEN M4=L(I)
9160 NEXT I
9200 PRINT:PRINT: " ¿QUIERES ESCOGER LAS DOS"
9210 PRINT " PRIMERAS NOTAS ? (S/N) ";
9220 INPUT R$:IF R$<>"S" AND R$<>"N" THEN 9200
9230 IF R$="N" THEN 9300
9240 PRINT:PRINT " PRIMERA NOTA : (";M1;"-";M2;") ";
9245 INPUT PN:IF PN<M1 OR PN>M2 THEN 9240
9250 PRINT " DURACION :  (";M3;"-";M4;") ";
9255 INPUT PD:IF PD<M3 OR PD>M4 THEN 9250
9260 PRINT:PRINT " SEGUNDA NOTA : (";M1;"-";M2;") ";
9265 INPUT SN:IF SN<M1 OR SN>M2 THEN 9260
9270 PRINT " DURACION :  (";M3;"-";M4;") ";
9275 INPUT SD:IF SD<M3 OR SD>M4 THEN 9270
9290 GOTO 9400
9300 PN=M1+INT(RND(1)*(M2+1-M1)):F1=0
9310 FOR I=1 TO MAX
9315 IF PN=N(I) THEN F1=1:I=MAX
```



```
9320 NEXT I
9325 IF F1=0 THE 9300
9330 PD=M3+INT(RND(1)*(M4+1-M3)):F1=0
9335 FOR I=1 TO MAX
9340 IF PF=L(I) THEN F1=1:I=MAX
9345 NEXT I
9346 IF F1=0 THEN 9330
9348 SN=M1+INT(RND(1)*(M2+1-M1)):F1=0
9350 FOR I=1 TO MAX
9355 IF SN=N(I) THEN F1=1:I=MAX
9360 NEXT I
9365 IF F1=0 THEN 9350
9370 SD=M3+INT(RND(1)*(M4+1-M3)):F1=0
9375 FOR I=1 TO MAX
9380 IF SD=L(I) THEN F1=1:I=MAX
9385 NEXT I
9388 IF F1=0 THEN 9370
9400 NN(1)=PN:NN(2)=SN:LL(1)=PD:LL(2)=SD
9405 PRINT:PRINT:PRINT "  ESTOY TRABAJANDO .... "
9410 FOR I=3 TO LP
9412 FOR J=0 TO 96:EN(J)=0:EL(J)=0:NEXT J:NM=0
9415 N1=NN(I-2):N2=NN(I-1):L1=LL(I-2):L2=LL(I-1)
9420 FOR J=1 TO MAX-2
9430 IF N(J)<>N1 THEN 9460
9440 IF N(J+1)<>N2 THEN 9460
9450 EN(N(J+2))=EN(N(J+2))+1:NM=NM+1
9460 NEXT J
9462 IF NM<>0 THEN 9470
9464 NN(I)=INT((N1+N2)/2+RND(1)*7-3)
9466 IF NN(I)<M1 OR NN(I)>M2 THEN 9464
9468 GOTO 9500
9470 NS=INT(RND(1)*NM+1):SN=0
9480 FOR J=0 TO 96
9485 SN=SN+EN(J)
9490 IF SN>=NS THEN NN(I)=J:J=96
9495 NEXT J
9500 NM=0
9530 IF L(J)<>L1 THEN 9560
9540 IF L(J+1)<>L2 THEN 9560
9550 EL(L(J+2))=EL(L(J+2))+1:NM=NM+1
9560 NEXT J
9562 IF NM<>0 THEN 9570
9564 LL(I)=INT((L1+L2)/2+RND(1)*7-3)
9566 IF LL(I)<M3 OR NN(I)>M4 THEN 9564
9568 GOTO 9569
9570 LS=INT(RND(1)*NM+1):SL=0
9580 FOR J=0 TO 96
9585 SL=SL+EL(J)
9590 IF SL>=LS THEN LL(I)=J:J=96
9595 NEXT J
9596 NEXT I
9599 GOTO 2000
```

Creando una melodía

Vamos a examinar ahora la otra posibilidad de la que hablábamos: la composición de una pieza musical siguiendo las reglas de un tipo específico de música.

El campo es muy amplio: puede haber tantos programas como géneros musicales existen. Es más, para cada uno de ellos se pueden escribir docenas de variantes: podemos seguir una gran serie de reglas vinculantes (con resultados poco variados y originales) o reglas más relajadas (con resultados tal vez demasiado aleatorios); hay mucho donde elegir. Vamos a contentarnos con un pequeño programa que puede servir como base para otros de este tipo.

Decidimos crear una pieza monofónica de estilo no muy definido, con una estructura rítmica clásica y una línea melódica que obedece a los principales cánones de la música tonal. Este podría ser el soporte melódico para una canción que, tendremos que reconocerlo, no será jamás el éxito del verano que viene (ni de los siguientes...)

He aquí las reglas que nos hemos impuesto:

1) La pieza se divide en tres partes, cada una con una longitud de ocho compases (podemos escoger entre compás de 3/4 y compás de 4/4).
2) Estructura rítmica: hay cinco esquemas rítmicos disponibles para cada medida (están indicados en las líneas DATA 21000-22040 con el mismo código utilizado para la duración de las notas: la línea 21040 contiene los números 16, 4, 4 seguidos de ceros, lo que corresponde a un compás compuesto por una blanca seguida de dos corcheas). La estructura rítmica de las cuatro primeras medidas de cada parte se escoge aleatoriamente y se repite en las cuatro siguientes. Además, la estructura rítmica de la tercera parte es idéntica a la de la primera.
3) Estructura melódica: se utilizan sólo las notas de la escala en DO mayor según los códigos especificados en las líneas 20000-20070 (al cambiar estas líneas de DATA podemos modificar la tonalidad de la pieza).

La primera nota de cada medida se escoge por "sorteo" conducido a partir de la primera nota de la pieza según la siguiente tabla:

Separación con la primera nota	Probabilidades
0	2/8
1	3/8
2	2/8
3	1/8

La separación, naturalmente, se cuenta dentro de la escala utilizada partiendo del DO (una separación igual a 1 produce un RE, una de 3 produce un FA).

Las otras notas se escogen a partir de la nota precedente siguiendo una regla análoga, pero con distintas posibilidades.

Separación con la nota precedente	Probabilidades
0	3/8
1	2/8
2	1/8
*	2/8

El caso señalado con "*" indica que la nota elegida es igual a la primera nota del compás.

En esta segunda tabla la separación también puede ser negativa: para la segunda nota del teclado el signo de separación es arbitrario, pero a continuación la elección se hace de forma que haya tres posibilidades sobre cuatro de que el sentido de la variación de altura sea el mismo que el observado para las dos notas anteriores. El objetivo es el de privilegiar secuencias de notas ascendentes o descendentes.

Además, las notas de los dos primeros compases de la tercera parte son idénticas a las correspondientes de la primera parte.

4) Ultima regla: cada parte termina con una pausa y la última nota de cada parte tiene el mismo tono que la primera.

Está claro que este conjunto de reglas es sólo uno de los tantísimos que podemos imponer. Pero hay aquí unas pocas ideas-clave que habrá que respetar si queremos obtener composiciones con un mínimo de armonía:

- Una estructura rítmica regular;
- notas que pertenezcan a una escala determinada (si queremos producir música tonal, naturalmente);
- línea melódica sin grandes saltos, con muchas secuencias ascendentes y descendentes y repeticiones de notas de base;
- repetición de esquemas, tanto melódicos como rítmicos.

Otra idea que no hemos aprovechado en el programa tiene que ver con la estructura armónica: deberíamos dar preferencia dentro de un compás a notas que pertenecen a un acorde determinado.

Otras propuestas para mejorar el proyecto (a costa, naturalmente, de posteriores esfuerzos de programación) serían: resaltar ciertas notas según su posición en un compás y utilizar para ellas un algoritmo de creación diferente; podríamos también favorecer la aparición de secuencias formadas por dos notas cercanas entre sí, seguidas de una serie de notas que cubran el intervalo entre las dos (este tipo de secuencia es muy frecuente en las melodías tradicionales), y todas las reglas que se quiera...

Tal y como está el programa produce composiciones que no resultan desagradables, pero que no poseen ni personalidad ni homogeneidad. De vez en cuando surge una melodía más lograda, pero no podemos decir que sea del tipo que motive a silbarla espontáneamente...

Uso del programa

Al comienzo hay que escoger el tipo de compás (3/4 ó 4/4) y la primera nota (entre las autorizadas) siguiendo el mismo código.

Bastará con esperar un poco para que, tras unas décimas de segundo, podamos escuchar la inolvidable composición creada por el ordenador...

Modificaciones

Hay miles de formas de alterar el proceso de creación de la pieza. Empezaremos por las modificaciones más inmediatas. Para cambiar los esquemas rítmicos hay que modificar las líneas 21000-21040 (compás 3/4) ó 22000-22040 (compás 4/4). Es posible también aumentar el número de esquemas al cambiar los valores 40 y 80 en las líneas 1010, 1210 y 2140, pero no es muy aconsejable para no agravar el defecto de homogeneidad de la pieza.

Al alterar las líneas 20000-20070, en cambio, se modifican las notas a nuestra disposición. No es obligatorio limitarse a las siete notas de una escala diatónica, pero sin exagerar si queremos conservar un carácter tonal en la composición.

Otra modificación elemental consiste en cambiar las probabilidades de aparición de las distintas notas: líneas 810-830 para la primera nota de un compás, líneas 510-530 para las siguientes.

Será necesario hacer alguna reestructuración del programa si

queremos cambiar otros parámetros: longitud de la pieza, frecuencia y ubicación de las repeticiones, etc.

Pueden utilizarse las otras reglas que hemos mencionado anteriormente o todas aquellas que le pasen por la mente; tendremos sólo el problema de la elección...

Este programa puede ponerse en marcha rápidamente, pero sigue un esquema de composición bastante rígido. Podemos modificarlo (sería mejor escribirlo de nuevo) para composiciones más libres, pero habría que incluir numerosos parámetros en la fase inicial (longitud de la pieza, repeticiones, probabilidad de aparición de las notas...). Podemos también añadir al programa una subrutina de salvaguarda de la obra creada.

El hecho de crear música polifónica es todo un caso aparte: habría que estudiar las reglas de la armonía y prepararse para un trabajo de programación algo más largo.

Descripción del programa

Para una buena comprensión del programa hay que tener bien claras las reglas de composición utilizadas.

500-790: Subrutina de elección de las notas en el interior de un compás.
500-600: Elección aleatoria de la separación con la nota precedente.
610-630: Si no es la segunda nota del compás, elección del signo de la separación.
640-650: Cálculo de la separación algebraica.
700-790: Cálculo del código de la nota correspondiente.
800-890: Subrutina de elección de la primera nota de un compás.
800-840: Elección casual de la selección.
850-890: Nota correspondiente.
1000-1230: Inicialización. Lectura de los vectores.
2000-2220: Introducción del tipo de compás y de la primera nota.
3000-3030: Comprobación: la nota elegida debe formar parte de las notas válidas.
4000-4010: Inicialización de las variabales (I = índice de la nota creada).
5000: Inicio del bucle de construcción de las dos primeras partes.
5100: Inicio del bucle de construcción de un compás.
5200-5220: Elección casual de uno de los cinco esquemas rítmicos posibles.

5300-5340: Atribución de la duración de la nota en curso según el esquema rítmico elegido.
5400: Llamada a la subrutina de elección de la altura de la nota.
5700-5810: Finalización de los dos bucles.
6100: Inicio del bucle de creación de la tercera parte de la pieza.
6200-6320: Cálculo de la duración de la nota (utilizando el esquema rítmico de la primera parte de la pieza).
6325-6400: Elección de la altura de la nota.
6700-6710: Finalización del bucle.
8000-8030: Interpretación de la pieza.
20000-20070: Códigos de las notas autorizadas.
21000-21040: Esquemas rítmicos (compases 3/4).
22000-22040: Esquemas rítmicos (compases 4/4).

LISTADO 6.2

```
1 GOTO 1000
500 RN=INT(8*RND(1))
510 IF RN<3 THEN SN=0:GOTO 600
520 IF RN<5 THEN SN=1:GOTO 600
530 IF RN<6 THEN SN=2:GOTO 600
540 N(I)=N2:GOTO 790
600 IF NR=2 THEN 640
610 SC=N(I-1)-N(I-2):SC=SGN(SC):IF SC=0 THEN 640
620 RS=INT(2*RND(1))
630 IF RS=0 THEN SN=SN*SC:GOTO 700
640 RS=INT(2*RND(1))
650 IF RS=0 THEN SN=-SN
700 FOR K=1 TO 55
710 IF N(I-1)=NN(K) THEN KN=K:K=55
720 NEXT K
730 N(I)=NN(KN+SN)
790 RETURN
800 IF J=I THEN N(I)=N1:GOTO 890
805 RN=INT(7*RND(1))
810 IF RN<2 THEN SN=0:GOTO 850
820 IF RN<5 THEN SN=1:GOTO 850
830 IF RN<6 THEN SN=2:GOTO 850
840 SN=3
850 N(I)=NN(K1+SN):N2=N(I)
890 RETURN
1000 GOSUB 10000
1010 DIM NN(55),RT(80)
1020 DI L(200),N(200)
1100 FOR I=1 TO 55
1110 READ NN(I)
1120 NEXT I
1200 MR=80:TT=0:K1=0
1210 FOR I=1 TO 80
1220 READ RT(I)
1230 NEXT I
2000 GOSUB 40
```



```
2100 PRINT:PRINT "     TIPO DE BATUTA  (3-4) : ";
2110 INPUT TP
2120 IF TP<>3 AND TP<>4 THEN 2100
2140 IF TP=4 THEN TT=40
2200 PRINT:PRINT "     PRIMERA NOTA (24-47) : ";
2210 INPUT N1
2220 IF N1<24 OR N1>47 THEN 2200
3000 FOR I=1 TO 55
3010 IF N1=NN(I) THEN K1=I:K=55
3020 NEXT I
3030 IF K1=0 THEN 2200
4000 I=1
4010 N2=N1
5000 FOR J1=1 TO 2
5100 FOR J=1 TO 8
5110 NR=1
5200 IF J>4 THEN R(J)=R(J-4):GOTO 5300
5220 R(J)=INT(5*RND(1))
5300 RR=8*R(J)+NR+TT
5310 IF RT(RR)=0 THEN 5705
5320 L(I)=RT(RR)
5340 IF NR=1 THEN GOSUB 800:GOTO 5700
5400 GOSUB 500
5700 I=I+1:NR=NR+1:GOTO 5300
5705 IF J1=1 THEN R1(J)=R(J)
5710 NEXT J
5720 N(I-1)=0:N(I-2)=N1
5800 NEXT J1
5810 I1=I-1
6100 FOR J=1 TO 8
6110 NR=1
6200 R(J)=R1(J)
6300 RR=8*R(J)+NR+TT
6310 IF RT(RR)=0 THEN 6705
6320 L(I)=RT(RR)
6325 IF J<3 THEN N(I)=N(I-I1):GOTO 6700
6340 IF NR=1 THEN GOSUB 800:GOTO 6700
6400 GOSUB 500
6700 I=I+1:NR=NR+1:GOTO 6300
6705 NEXT J
6710 N(I-1)=0
8000 FOR I=1 TO 200
8005 IF N(I)+N(I+1)=0 THEN I=200:GOTO 8030
8010 N=N(I):L=L(I)
8020 GOSUB 10
8030 NEXT
9000 END
20000 DATA 2,4,5,7,9,11
20010 DATA 12,14,16,17,19,21,23
20020 DATA 24,26,28,29,31,33,35
20030 DATA 36,38,40,41,43,45,47
20040 DATA 48,50,52,53,55,57,59
20050 DATA 60,62,64,65,67,69,71
20060 DATA 72,74,76,77,79,81,83
20070 DATA 84,86,88,89,91,93,95
21000 DATA 8,8,8,0,0,0,0,0
21010 DATA 16,8,0,0,0,0,0,0
```

```
21020 DATA 8,8,4,4,0,0,0,0
21030 DATA 4,4,4,4,8,0,0,0
21040 DATA 16,4,4,0,0,0,0,0
21050 DATA 8,8,8,8,0,0,0,0
22000 DATA 8,8,8,8,0,0,0,0
22010 DATA 16,16,0,0,0,0,0,0
22020 DATA 16,8,8,0,0,0,0,0
22030 DATA 8,4,4,8,4,4,0,0
22040 DATA 8,4,4,16,0,0,0,0
```


CAPITULO VII

EL ORDENADOR COMO COMPOSITOR (MENOS CONVENCIONAL)

Como ya prometimos, trataremos aquí de utilizar el ordenador para crear música de tipo menos clásico que la producida por los programas del capítulo anterior. Entramos ahora en un universo muy amplio, todavía no del todo explorado, del que podremos obtener grandes satisfacciones.

Si las creaciones de música computerizada de tipo convencional han resultado ser bastante decepcionantes, en el campo de la experimentación el balance es más positivo, con producciones a veces muy satisfactorias. Bueno, tal vez no todos piensen igual y hay quien no aguantará un tipo de música que se sale de los esquemas culturales tradicionales. Está claro que se requiere cierta disponibilidad e incluso la voluntad de modificar las costumbres de audición. La "otra" música no compite ni quiere compararse con la tradicional: se trata de algo diferente y no podemos juzgarla según los mismos criterios.

Tal vez se esté haciendo una pregunta de este tipo: el uso del ordenador ha ayudado, sin duda alguna, a obtener brillantes resultados, pero esto ha sido en estudios en donde se dispone de un material muy sofisticado respecto del cual un micro doméstico causa risa, ¿merece la pena, entonces, intentarlo? Nuestra respuesta es afirmativa con tal de que actúe con cierta modestia: un Commodore 64 no puede dar resultados comparables con los obtenidos por el 4X de la firma IRCAM, eso está claro; pero dentro de sus limitaciones, los mejores generadores musicales en circulación dejan un amplio margen para la experimentación, y en este caso lo fundamental es la idea (cosa que no suele faltar precisamente).

Pasemos ahora a la práctica. Tan sólo un comentario antes de seguir adelante: este capítulo contiene muchas ideas y propuestas que hay que profundizar, pero pocos programas. El motivo es sencillo: una vez que nos salimos de las producciones corrientes se hace necesario conocer y aprovechar al máximo las características particulares del generador sonoro de que se disponga. Esto quiere decir que hay que escribir programas enteramente dedicados a un modelo específico; aún más: se descubren rápidamente las limitaciones del BASIC y no es posible obtener muchos efectos sin recurrir al lenguaje máquina. De manera que debemos retornar al manual de referencia del ordenador para conocer todos los detalles del generador sonoro. Tal vez haya llegado también el momento de empezar a programar en lenguaje máquina, aunque en un principio basta con que sean sencillas subrutinas integradas en un programa BASIC, y darle luego tiempo al tiempo...

Debería ya haber quedado claro que con nuestros programas, aplicables a un generador sonoro "mínimo", no es posible llegar muy lejos, pero veamos ahora una primera aplicación bastante inmediata.

Minieditor musical (de nuevo)

A pesar de que el título (y la estructura general del programa) recuerda el editor del capítulo 5, la idea es totalmente diferente. Se trata también en este caso de introducir una pieza musical en la memoria y hacerla sonar, pero la construcción de la pieza es mucho más libre.

La pieza musical es considerada también como una sucesión de "notas". Cada "nota" posee una duración (expresada en ochentavos de segundo), pero el sistema utilizado para fijar el tono es diferente. Nadie nos obliga a utilizar las alturas presentes en la escala de siempre. Podemos indicar una frecuencia cualquiera (expresada en Hz) siempre que, naturalmente, esté dentro de los límites audibles y de los del ordenador.

La elección no se limita a eso: la frecuencia no tiene por qué ser fija para toda la duración de la nota; basta con indicar dos frecuencias diferentes y la nota se convierte en un sonido que pasa regularmente de la primera frecuencia a la segunda en la duración elegida con anterioridad.

Esto, por desgracia, es sólo en teoría, ya que pronto empezamos a descubrir los límites del BASIC: lo que debería ser un sonido continuado de altura variable se transforma en una sucesión de sonidos breves de altura fija. La interrupción entre cada fracción de nota se hace manifiesta debido a la lentitud del intérprete

BASIC que gestiona la sucesión de estos sonidos. Este será el primer punto que habrá que mejorar cuando adaptemos el programa a un ordenador específico.

Las otras opciones son parecidas a las del editor del capítulo 5.

Uso del programa

Damos por supuesto que posee un mínimo de práctica con el programa del listado 5.2 y es por eso que pasaremos velozmente por las funciones idénticas en los dos programas.

Tras haber metido en memoria el programa del listado 7.1 y haberlo salvado en cinta o disco podemos apretar el RUN.

El menú principal presenta las siete opciones de siempre:

1) CREAR MUSICA

El programa imprime el número de línea (una para cada nota) y solicita seguidamente las dos frecuencias y la duración de la nota. Las duraciones son en ochentavos de segundo y las frecuencias en Hz (las frecuencias audibles van, aproximadamente, desde 20 Hz a 18.000 Hz, pero muchos ordenadores son más limitados). Contrariamente a lo que ocurre para el editor del capítulo 5, hay que pulsar RETURN después de cada valor.

Para obtener una nota de frecuencia fija bastará con indicar una segunda frecuencia igual a la primera, o simplemente apretar el "espacio" como respuesta para esta segunda frecuencia. En cambio, al responder espacio a la petición de longitud estamos fijando una duración igual a la de la nota anterior.

Para incluir una pausa en la pieza es suficiente con introducir una nota de la duración deseada y de frecuencia exterior al intervalo audible.

Al final de cada línea hay que pulsar RETURN para pasar a la siguiente nota o apretar la tecla "R" para meter de nuevo los datos de esa misma nota.

Para terminar la introducción de las notas hay que teclear "ESC" al comienzo de una línea.

2) LISTADO DE LA PIEZA

Idéntico a la opción análoga del programa 5.2.

3) MODIFICACION DE LA PIEZA

Pasa lo mismo. Tras haber fijado la primera línea que hay que modificar es necesario introducir los datos como en la opción 1).

4) INTERPRETAR MUSICA

Una opción que puede reservar algunas sorpresas...

5) y 6) SALVAR Y CARGAR UNA PIEZA

Igual que en el programa 5.2, estas opciones deben ser programadas en función del ordenador y periféricos disponibles.

Hay una diferencia esencial respecto del programa anterior: se trata, en esta ocasión, de memorizar no ya dos, sino tres vectores unidimensionales: F1, F2 y L. No olvide salvar (y restaurar) la longitud de la pieza, MAX, en el elemento 0 de cada vector.

7) FIN

Modificaciones

La primera modificación, como ya dijimos, afecta a la subrutina que hace sonar una nota de frecuencia variable. El hecho de escribir una subrutina en lenguaje máquina resolvería el problema, pero tal vez no sea siquiera necesario. En muchos ordenadores hay un registro (o dos) del generador sonoro que contienen la frecuencia de la nota. Basta con dar a la tecla que provoca la emisión de una nota de frecuencia FR1 para la duración TOTAL de la nota, y durante la emisión modificar en tiempo real el contenido del registro de frecuencia. Hay que hacer alguna prueba para hallar la temporización optima. La cosa resulta mucho más sencilla si el BASIC dispone de una instrucción del tipo ON INTERVALL...GOSUB (igual que en el BASIC MSX) que permite gestionar las interrupciones sin programar en lenguaje máquina.

Puede aumentarse el número y funciones de los parámetros a disposición de cada nota. Aunque los únicos controles que dispongamos sean los de frecuencia y duración, podemos añadir un parámetro para controlar la velocidad de variación de la frecuencia; esta variación puede ser logarítmica en lugar de lineal (lenta al comienzo de la nota y rápida al final, o al contrario).

También podríamos añadir, si fuera posible, el control del volumen: cada nota con un parámetro para indicar su volumen, o mejor dos parámetros si queremos que suenen con un volumen variable (igual que para la frecuencia). Un tercer parámetro podría indicar la velocidad y la forma (lineal o no) de esta variación. Así es posible controlar libremente la envolvente del sonido.

Nadie nos impide añadir otros parámetros para una posible elección de la forma de onda o de otras características.

Al aumentar el número de parámetros se pone cada vez más

en evidencia la lentitud del BASIC, por lo que se va haciendo necesario pasar al lenguaje máquina o, al menos, a un control más directo del generador sonoro. Se requerirá también modificar la presentación en la pantalla (los parámetros de una nota no podrán ya ocupar sólo una línea).

Las modificaciones citadas a propósito del programa 5.2 son posibles también aquí, en concreto la mejora de la seguridad en la fase de introducción.

Descripción del programa

100-190: Subrutinas de introducción de los datos durante las opciones 1) y 3). No hemos utilizado la instrucción INPUT para no molestar la presentación del listado en la pantalla (en particular para evitar el salto de línea y después de cada parámetro). La cadena que se da por respuesta se construye carácter por carácter con la instrucción INKEY$ (o GET). Si el carácter recibido es un "RETURN" (código ASCII 13) se sale de la subrutina. Nótese que INKEY$ y GET no imprimen el carácter recibido. Es necesario añadir la instrucción PRINT R$ para controlar el valor introducido.

1000-1110: Inicialización.

2000-2230: Menú principal.

3000: Inicio de la subrutina de introducción de los datos.

3100-3130: Se comprueba que la pieza no haya superado la longitud autorizada (1000 notas en el listado publicado, aptas para ser ampliadas si se desea).

3200-3250: Introducción de la primera frecuencia. Tras haber comprobado que el primer carácter no es "ESC" (en este caso se salta a 3900), el valor solicitado se introduce como cadena con la ayuda de la subrutina de la línea 100. La respuesta se traduce luego en número y se imprime de nuevo para un posterior control.

3300-3340: Igual procedimiento para la introducción de la segunda frecuencia. Si la respuesta es "espacio" (RR$=""), la segunda frecuencia toma un valor igual al de la primera.

3400-3440: Lo mismo vale para la duración. En caso de respuesta "espacio" la duración de la nota será igual a la de la nota anterior.

3500-3600: Fin de introducción de una nota. Si se pulsa "R" pueden meterse de nuevo los datos de la misma nota.

3900-3920: Salida de la subrutina de construcción de una pieza.

4000: Inicio de la subrutina que interpreta la pieza en memoria.

4100-4140: Si las dos frecuencias son iguales suena la nota con la frecuencia común. El multiplicador de la línea 4120 (L=L(I)*5)

sirve para compensar el tiempo empleado por el BASIC para sus cálculos en el caso de frecuencias diferentes. Hay que adoptar el valor para cada modelo de ordenador. Si incluimos una subrutina musical más eficaz, este multiplicador deberá ser suprimido.

4200-4230: Si las dos frecuencias son distintas sonarán varias notas de duración mínima y frecuencias intermedias entre las dos elegidas.

4400: Fin de la subrutina de emisión musical.

5000: Inicio de la subrutina que da el listado de la pieza en memoria.

5100-5130: Comprueba que hay efectivamente una pieza en la memoria.

5200-5240: Introducción de las líneas iniciales y finales que hay que listar y control de la validez de estos datos.

5300-5520: Se imprimen los valores de los parámetros para cada nota.

5600-5610: Salida de la subrutina.

6000: Inicio de la subrutina de modificación de una pieza.

6010-6040: Control de la presencia de una pieza en memoria.

6100-6130: Tras la introducción de la línea inicial el flujo del programa se desvía hacia la subrutina de creación de una pieza. La bandera FL igual a 1 indica que estamos en la opción 3.

7000-8000: Se incluyen aquí las subrutinas de salvaguarda y carga de la composición.

LISTADO 7.1

```
1       GOTO 1000
100     RR$=""
110     R$=INKEY$: IF R$="" THEN 110
120     IF ASC(R$) = 13 THEN 190
130     RR$=RR$+R$: PRINT R$;
140     GOTO 110
190     RETURN
1000    GOSUB 10000
1100    DIM F1(1000),F2(1000),L(1000)
1110    MAX=0 : F=0
2000    GOSUB 40
2100    PRINT : PRINT : PRINT: PRINT
2110    PRINT " 1 = CREAR MUSICA"
2120    PRINT " 2 = LISTADO DE LA PIEZA"
2130    PRINT " 3 = MODIFICACION DE LA PIEZA"
2140    PRINT " 4 = INTERPRETAR MUSICA"
2150    PRINT " 5 = SALVAR UNA PIEZA"
2160    PRINT " 6 = CARGAR UNA PIEZA"
2170    PRINT " 7 = FIN"
2180    PRINT
2200    INPUT R$
2210    R=VAL(R$)
```



```
2220    IF R<1 OR R>7 THEN 2000
2230    ON R GOTO 3000, 5000, 6000, 7000, 8000, 2500
2500    END
3000    GOSUB 40
3010    I=0 : FL=0
3100    PRINT : I=I+1 : IF I<1001 THEN 3200
3110    PRINT "MEMORIA AGOTADA"
3120    GET R$
3130    GOTO 2000
3200    X=1 : GOSUB 50
3210    PRINT I; : X=5 : GOSUB 50 : PRINT "FR1 = ";
3220    GOSUB 100
3230    IF ASC(RR$)=27 THEN 3900
3240    F1(I)=VAL(RR$)
3250    X=9 : GOSUB 50 : PRINT F1(I); : PRINT " ";
3300    X=15 : GOSUB 50
3310    PRINT "FR2 = ";
3315    GOSUB 100
3320    IF RR$=" " THEN F2(I)=F1(I): GOTO 3340
3330    F2(I)=VAL(RR$)
3340    X=19 : GOSUB 50 : PRINT F2(I); : PRINT " ";
3400    X=25 : GOSUB 50
3410    PRINT "DUR = ";
3415    GOSUB 100
3420    IF RR$=" " THEN L(I)=L(I-1) : GOTO 3440
3430    L(I)=VAL(RR$)
3440    X=29 : GOSUB 50 : PRINT L(I); : PRINT " ";
3500    GET R$
3510    R=ASC(R$)
3520    IF R=82 THEN 3200
3530    IF R<>13 THEN 3500
3600    GOTO 3100
3900    IF FL=0 THEN MAX=I-1 : GOTO 2000
3910    IF I-1>MAX THEN MAX=I-1
3920    GOTO 2000
4000    GOSUB 40
4100    FOR I=1 TO MAX
4110    IF (F1(I)<>F2(I)) AND (L(I)>1) THEN 4200
4120    F=F1(I) : L=L(I)*5
4130    GOSUB 60
4000    GOSUB 40
4100    FOR I=1 TO MAX
4110    IF (F1(I)<>F2(I)) AND (L(I)>1) THEN 4200
4120    F=F1(I) : L=L(I)*5
4130    GOSUB 60
4140    GOTO 4300
4200    LL=L(I)-1 : L=1 : DF=INT((F2(I)-F1(I))/LL)
4210    FOR J=0 TO LL
4220    F=F1(I)+DF*J : GOSUB 60
4230    NEXT J
4300    NEXT I
4400    GOTO 2000
5000    GOSUB 40
5100    IF MAX<>0 THEN 5200
5110    PRINT "         MO HAY NINGUNA"
5120    PRINT "    PIEZA EN MI MEMORIA"
5125    GET R$
```

```
5130    GOTO 2000
5200    PRINT "LINEA INICIAL";
5210    INPUT R$ : R1=VAL(R$)
5212    IF R1<1 OR R1>MAX THEN GOSUB 40 : GOTO 5200
5220    PRINT "LINEA FINAL"
5230    INPUT R$ : R2=VAL(R$)
5232    IF R$=" " THEN R2=MAX
5234    IF R2<1 OR R2>MAX THEN GOSUB 40 : GOTO 5220
5240    PRINT
5300    FOR I=R1 TO R2
5310    PRINT I;
5320    X=5; GOSUB 50
5330    PRINT "FR1 = "; : PRINT F1(I);
5400    X=15 : GOSUB 50
5410    PRINT "FR2 = "; : PRINT F2(I);
5500    X=25 : GOSUB 50
5510    PRINT "DUR = "; : PRINT L(I)
5520    NEXT I
5600    GET R$
5610    GOTO 2000
6000    GOSUB 40
6010    IF MAX<>0 THEN 6100
6020    PRINT "        NO HAY NINGUNA"
6030    PRINT "   PIEZA EN MI MEMORIA"
6035    GET R$
6040    GOTO 2000
6100    PRINT "LINEA INICIAL";
6110    INPUT R$ : R=VAL(R$)
6120    IF R<1 OR R>MAX+1 THEN GOSUB 40 : GOTO 6100
6130    I=R-1 : FL=1 : GOTO 3100
7000    GOTO 2000
8000    GOTO 2000
```

¿Y ahora?

Bueno, ha llegado el momento de dar rienda suelta a nuestra imaginación. El ordenador nos da una gran libertad y en nuestras manos está el aprovecharnos: nada nos impide tratar de poner en práctica nuestras más extrañas ideas, es decir, JUGAR (no es totalmente casual que muchos idiomas utilicen una única palabra para decir "jugar" y "tocar un instrumento"...).

¿Por dónde empezar? Antes que nada veamos los parámetros que tenemos a nuestra disposición y cómo actuar sobre ellos.

1) FRECUENCIA

El ordenador nos permite escoger con gran libertad la frecuencia del sonido emitido, lo que es una gran ventaja. La música tradicional debe contentarse con utilizar notas de alturas fijas dentro de la escala musical elegida. La historia de la música de este siglo se caracteriza por el deseo de actuar con más libertad a tra-

vés de distintas experiencias (atonalidad, dodecafonía...), pero los límites fijos de gran parte de los instrumentos no permitían salirse de la partición clásica de la octava en doce semitonos. No tenemos nada en contra de la escala clásica, justificada por razones no sólo históricas, sino también físicas (no tenemos espacio aquí para profundizar sobre el tema), pero resulta muy interesante probar algo diferente.

Echemos una ojeada al manual del ordenador para recordar cómo convertir la frecuencia deseada en parámetros aceptables por el ordenador (esta conversión suele ser muy sencilla) y pongámonos a trabajar. Si decidimos utilizar un número limitado de frecuencias conviene construir en memoria una tabla con los valores correspondientes para evitar perder el tiempo en el futuro debido a esta conversión.

Podemos decidir crear música según escalas más o menos extrañas utilizadas en otras épocas y lugares:

— escala pitagórica, antepasada de nuestra actual escala, con siete notas de frecuencias f, f*9/8, f*81/64, f*4/3, f*3/2, f*27/16, f*243/128;
— escala pentafónica, empleada desde China hasta Perú pasando por Escocia, que utiliza cinco notas cuyas frecuencias son f, f*9/8, f*81/64, f*3/2, f*27/16;
— y (¿por qué no?) la escala heptatónica de los indígenas "Are" de las Islas Salomón: se sirve de siete notas distribuidas regularmente en la octava (la frecuencia de una nota se obtiene al multiplicar la de la precedente por la raíz séptima de dos).

Naturalmente podemos inventar nuestra escala personal. Podemos dividir la octava en un número de intervalos regulares distintos de los corrientes; en estos casos hay que multiplicar las frecuencias sucesivas por las raíces seis, diez o siete de dos. El asunto empieza a resultar interesante al tratar de crear acordes con nuestras nuevas notas...

Vamos a dar un paso adelante y eliminar cualquier sistema de referencia para utilizar todo el intervalo de frecuencias a nuestra disposición. Podemos obtener efectos espectaculares al pasar constantemente de una frecuencia a otra o tocando notas cuya frecuencia se aleja insensiblemente de la de partida. Esto es aún mejor en una pieza polifónica con evoluciones conjuntas de distintas notas emitidas a un tiempo. El abanico de posibilidades es amplísimo, pero a menudo las disonancias que resultan se hacen insoportables para el menos convencional de los oyentes.

2) VOLUMEN

Las variaciones de volumen tienen gran importancia dentro del proceso de interpretación del sonido por parte de nuestro cerebro, pues sirven para caracterizar en parte el timbre de un instrumento, como ya vimos en el primer capítulo.

Si nuestro procesador sonoro dispone de un generador de envolvente podremos producir con facilidad sonidos de volumen variable en el tiempo. Si queremos algo más podemos tomar directamente el control del volumen y crear por nosotros mismos envolventes más extrañas.

Nadie ha dicho que una envolvente tenga por fuerza que contar con las cuatro fases clásicas del esquema ADSR: construyamos una envolvente de seis o diez segmentos; podemos alternar fases de disminución y alza del volumen sonoro, o crear velocísimos cambios de volumen con resultados bastante extraños.

Sin embargo, el control del volumen en la mayor parte de los ordenadores deja mucho que desear en lo relacionado con la definición. El volumen se especifica a menudo sobre cuatro bits (e incluso menos) y puede, por tanto, tomar como máximo dieciséis valores distintos. Esto es algo reducido para permitir ciertas florituras. Además, manipulaciones veloces del volumen (como también de la frecuencia) se hacen pronto imposibles en BASIC. Pero bueno, ya avisamos que antes o después habría que empezar con la programación en lenguaje máquina.

3) TIMBRE

El hecho de controlar el volumen es ya una primera manera de hacer lo mismo con el timbre de nuestro instrumento, pero para muchos generadores sonoros éste es el único método, ya que son muy pocos los que ponen a nuestra disposición distintas formas de onda. Esto es una pena, pues el hecho de jugar con las variaciones de timbre es una de las maneras más fascinantes de producir música con un ordenador. Una pieza musical puede ser construida para seguir una sucesión de timbres (y no sólo una línea melódica o armónica). Pueden obtenerse resultados inconcebibles sin el uso del ordenador.

Si el generador dispone de un número limitado de formas de onda (tipo Commodore 64) habrá que contentarse con saltar de una a otra produciendo efectos no muy convincentes. La posibilidad de modificar con continuidad el "pulso" de la onda rectangular puede dar resultados mejores.

Si la forma de onda está representada en RAM por una tabla (como en el Macintosh, el Amiga o el Music System) la variación de timbre puede producirse con la máxima libertad, pero requiere muchas transmisiones de datos en memoria. También en este caso la programación en lenguaje máquina se hace indispensable.

Los sistemas más favorecidos son aquellos en los que el sonido se genera en modulación de frencuencia. Basta con modificar poquísimos parámetros para obtener una forma de onda que se desarrolla en el tiempo sin otras intervenciones. En compensación, la elección de los parámetros necesarios para obtener el resultado deseado no es fácil (pero después de todo ¡el factor sorpresa forma parte del juego!).

Manejo de los parámetros

Ya hemos tratado de los parámetros que hay que controlar; digamos ahora dos palabras acerca de la forma de establecer la evolución de estos parámetros.

Una primera cuestión consiste en decidir sobre el papel (tras una posible experimentación) el esquema que debe seguir esta evolución. Los sucesivos valores pueden ser introducidos en las líneas DATA de un programa en BASIC o como tablas en un programa en lenguaje máquina.

A menudo es preferible establecer un algoritmo que guíe la evolución de cada parámetro partiendo de un conjunto más limitado de datos. Algoritmos posibles pueden encontrarse por miles, adoptados por distintas disciplinas. Las matemáticas son un pozo sin fondo a nuestra total disposición: ¡hay tantas funciones cuyos gráficos pueden crear graciosas melodías!

Sí, ya sabemos que más de uno pensará que una música producida de esa manera sólo puede ser fría y mecánica. Esto en parte es verdad, pero las matemáticas son más ricas de lo que generalmente pensamos y si buscamos más allá de las funciones corrientes podremos encontrar otras más "cálidas" y con un comportamiento poco "mecánico". Y nadie nos impide incluir un "ruido" en nuestro algoritmo para modificar de manera casual el desarrollo demasiado "serio" de nuestra función.

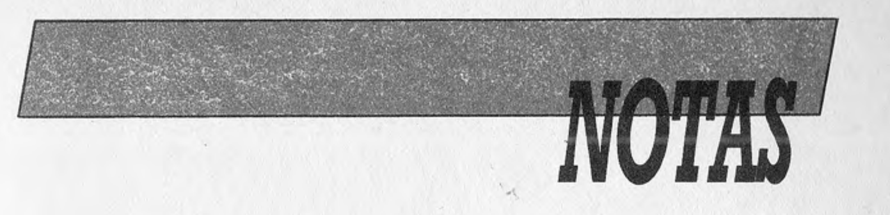
También podemos utilizar la estadística para nuestros fines, sin hablar de la física o la biología. ¿Por qué no programar una composición en la cual un parámetro se desarrolla como las moléculas de un líquido en ebullición o como una población animal en un ambiente ecológico?

Acabamos de hallar una nueva fuente para nuestra inspiración: la simulación. Hay tantos procesos que pueden hallar de alguna manera un equivalente musical... Esta es una manera de crear frecuente en la música tradicional, en donde situaciones reales son traducidas musicalmente por la mente del compositor. Lo mismo puede ocurrir utilizando el "cerebro" del ordenador, que, aun-

que inferior al humano, posee otras cualidades inequiparables. Si nuestro cerebro le comunica el programa necesario el ordenador podrá analizar un dibujo, por ejemplo, y transcribirlo musicalmente. Y, claro está, también es posible volver a elaborar una pieza de música tradicional...

Encontramos para los ordenadores más difundidos numerosos interfaces que comunican al aparato datos acerca del mundo exterior (señales visuales o sonoras, temperaturas, etc.). Pueden ser utilizados para transmitir elementos determinantes del proceso musical. También aparatos difundidísimos como el joystick o el ratón podrán ser utilizados para modificar en tiempo real el sonido procesado, introduciendo nuevamente en la música computerizada el factor humano, tan necesario.

Tan sólo hemos tocado el tema por encima; habría aún miles de ideas para poner en práctica, pero nuestra labor ha terminado y ha llegado su turno...





i bien es cierto que la potencia, flexibilidad, capacidades gráficas, etc., de un ordenador son características muy importantes a la hora de valorarlo, no lo es menos que en muchas aplicaciones la posibilidad de generar efectos sonoros o, incluso, música propiamente dicha, es imprescindible.

Desde los videojuegos hasta un uso profesional el campo a abarcar es inmenso y, desde luego, no todos los ordenadores son capaces de cubrirlo. En este libro veremos qué es eso de componer música y efectos sonoros con el ordenador, cómo lograrlo en cada caso y las posibilidades que nos ofrecen al respecto los distintos ordenadores presentes en el mercado.

395 pts.
(incluido IVA)